Preserving Inference Privacy in Multi-Agent Systems

Puspanjali Ghoshal

Preserving Inference Privacy in Multi-Agent Systems

Thesis submitted to

Indian Institute of Technology Guwahati

for the award of the degree

of

Doctor of Philosophy

by

Puspanjali Ghoshal

(Roll Number: 206123105)

under the guidance of

Prof. Ashok Singh Sairam

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

March 2026

Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati, Assam-781039, India

Declaration

It is certified that the work contained in this thesis entitled "**Preserving Inference Privacy in Multi-Agent Systems**" has been done by me, under the supervision of Dr. Ashok Singh Sairam, Professor, Department of Mathematics, Indian Institute of Technology Guwahati for the award of the degree of Doctor of Philosophy and this work has not been submitted elsewhere for a degree.

IIT Guwahati
March 2026

Puspanjali Ghoshal
Roll No. 206123105

Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati, Assam-781039, India

Certificate

This is to certify that this thesis entitled "**Preserving Inference Privacy in Multi-Agent Systems**" submitted by **Puspanjali Ghoshal**, to the Indian Institute of Technology Guwahati, is a record of bonafide research work carried out under my supervision and is worthy of consideration for award of the degree of Doctor of Philosophy of the Institute.

IIT Guwahati
March 2026

Dr. Ashok Singh Sairam
Professor
Department of Mathematics
Indian Institute of Technology Guwahati.

Dedicated to my beloved Ma and Babai.

For your endless love, quiet sacrifices, unwavering faith, and every dream you nurtured before I could dream for myself.

I love you.

Acknowledgment

It gives me tremendous pleasure and honor to be able to express my gratitude to all the people who have supported me in my PhD journey. I want to start by thanking Prof. Ashok Singh Sairam for supervising my thesis. Throughout my stay, his assistance, mentorship, and patience have served as an anchor for me. His never-ending perseverance, tolerance, and support have inspired me throughout. This thesis would not have been possible without his direction and insight.

Alongside my PhD supervisor, I would like to express utmost gratitude towards my doctoral committee members, Prof. Gautam Kumar Das, Prof. Partha Sarathi Mandal and Dr. Moumita Patra, for their constructive and insightful feedback during my PhD journey which has helped sharpen the thesis further.

I am ever so thankful to Prof. Kalpesh Kapoor and Prof. Jiten Kalita for always being there to listen to my problems, both curricular and non-curricular and for guiding me on how to handle any adverse situation that may arise during my academic journey and beyond. I am grateful to them for trusting me with the department representation responsibilities during the early years of my PhD.

The Indian Institute of Technology Guwahati deserves my heartfelt gratitude, specially the institute bus service, which made commuting effortless, specially to movie theaters and restaurants, when the occasional PhD stress needed a little escape.

I would like to thank Prof. Axel Sikora from Hochschule Offenburg, Germany for providing valuable suggestions during the DST-DAAD project tenure. Visiting the Offenburg campus with Prof. Sairam, during the project tenure, will forever remain as one of the highlights of my PhD journey.

I want to express my gratitude especially to Dr. Kuntal Banerjee and Dr. Shubhabrata Das, who have constantly supported, guided and believed in me since my master's. From writing countless reference letters to always pushing me to achieve the best I could in academia, they have been one of the key motivators who pushed me to aim for a PhD. Also, I would like to thank my high school mathematics teacher, Late Miss Shanti, whose encouragement and strict demeanor eventually helped develop an affinity for mathematics that shaped my future into what it is today.

I appreciate my research colleagues for creating an enjoyable workplace. I would like to specially mention my seniors, Ajay Kumar and Abhradeep Datta; and my juniors Sachin Sahu and Anshika Mittal for all the tea and evening snack sessions. Ajay da has been more of an elder brother than just a senior; always ready to help out with whatever issue I brought up to him.

I would like to thank my other set of friends and juniors from my PhD, Rajashree Bar, Adhiraj Mandal and Nilanjana Chowdhury. Much like Ajay da, Adhiraj has always been the little brother that I could vent to. I've known Rajashree for only around a year, but within that short span of time, she has managed to do a lot for me. The shared meals that we had while discussing the ups and downs of PhD life, will be cherished forever. Nilanjana has also been there throughout my journey as a pillar of support, while generously sharing her truckload of wisdom and occasional reality checks, whenever needed.

I am very grateful to Snigdha Bakshi who has been there for me since kindergarten. Over the years, our friendship has grown through countless memories, challenges, and moments of joy, and she has always stood by me with unwavering support, kindness and justified criticism whenever needed. She is an epitome of what true friendship looks like.

My heartfelt gratitude goes to Dr. Aayushman Raina for being a constant source of strength throughout this journey. As a trusted friend, confidant, and unwavering supporter, his quiet encouragement and understanding helped me navigate many challenging moments. I am deeply grateful for his companionship, patience, and belief in me, all of which made this journey more meaningful (special mention: the regular evening fast-food sessions!).

I wish to express my heartfelt remembrance of my maternal uncle, Late Deb Kumar Chakrabarti. He would have been immensely happy to see me earn my doctorate and would have proudly celebrated it. Though he is no longer with us, his encouragement and affection continue to live on in my memories.

Finally, I would like to express my deepest gratitude to my parents, Mr. Pulak Ghoshal and Mrs. Mitul Ghoshal. Whatever I am today is a reflection of their unconditional love, guidance, unwavering faith, and constant support. Their countless sacrifices and selfless efforts have enabled me to achieve every milestone I have ever dreamed of. They have always gone above and beyond to provide me with every opportunity to pursue my aspirations, placing my dreams before their own. Words cannot adequately express the depth of my gratitude or the extent of what I owe them. This degree belongs far more to them than it does to me.

And last but not the least, a quiet thank you to myself: for surviving reviewer comments, debugging inexplicable errors, rewriting chapters more times than I care to admit, and somehow making it to the end. Here's to the countless late nights and the determination that got me here.

March 2026 **Puspanjali Ghoshal**

Hakuna Matata

Abstract

A multi-agent system (MAS) involves autonomous agents observing and communicating with a fusion center, which aggregates the data to infer one or more system parameters with the highest possible accuracy. However, besides the system parameters, an honest-but-curious fusion center may infer secondary sensitive information or create a profile of an agent without its consent. These attacks constitute an inference privacy breach. This thesis aims to develop robust and efficient mechanisms to prevent such breaches in MASs. Since any entity other than the data owner can be a potential adversary, preserving inference privacy requires decentralized data sanitization. Moreover, the limited computational resources of autonomous agents necessitate lightweight sanitization mechanisms. As data sanitization inevitably modifies the raw observations and reduces utility, privacy mechanisms must preserve sufficient utility for accurate distributed inference at the fusion center.

The thesis investigates the applicability of differential privacy (DP) to mobile agents that continuously share their location information with a third party for location-based services. We show that the privacy guarantee provided by standard DP with fixed per-timestep budget allocation degrades over time due to sequential composition. Thus, we propose a perturbation generation mechanism that provides improved privacy preservation for continuous queries.

The thesis extends this framework to higher-dimensional data streams by proposing an adaptive privacy budget allocation scheme. The proposed method allocates the total available budget over timesteps according to the informativeness of the data and the importance of the agent to the system. Informative timesteps are prioritized to yield higher utility while maintaining cumulative privacy.

Next, the thesis moves on from perturbation-based methods to dimension reduction based methods. Exploiting the similarity between the MAS framework and robust concept deduction, a dynamic projection method is developed. The projection matrix evolves based on the previous raw observation and its sanitized counterpart, resulting in an adaptive data-dependent mechanism that achieves an effective privacy-utility tradeoff.

Finally, adaptive modulo hashing is introduced. The proposed method eliminates the requirement of a fixed divisor and produces data-dependent hash values for resource-constrained multi-agent systems. Collectively, the thesis proposes decentralized, lightweight, and computationally efficient solutions for preserving inference privacy in agent based distributed systems.

Abbreviation

AMHF	Adaptive Modulo Hash Function
APBA	Adaptive Privacy Budget Allocation
BRP	Basic Random Projection
CRLB	Cramér-Rao Lower Bound
DP	Differential Privacy
DPM	Dynamic Projection Method
FedAPCA (F-APCA)	Federated Adaptive Privacy-preserving Client-level Aggregation
FIM	Fisher Information Matrix
FL	Federated Learning
HE	Homomorphic Encryption
JL	Johnson–Lindenstrauss
LBS	Location based service
LDP	Local Differential Privacy
MAE	Mean Absolute Error
MAS	Multi-Agent Systems
MSE	Mean Squared Error
NRP	Neuronal Random Projection
PCA	Principal Component Analysis
S-DP	Shuffled DP
SGD	Stochastic Gradient Descent
TP	Trusted Party
VW	Variance weighted
WSN	Wireless Sensor Networks

Nomenclature

$\mathbf{x}$	System parameter
L	Dimension of system parameter
p_{jt} (or, p_j)	Private parameters
g_{jt} (or, g_j)	Public parameters
$MI(.;.)$	Mutual Information
$\mathcal{M}$	Privacy preserving mechanism
k (and k')	Number of private parameters in raw (and sanitized) data tuple
$\mathcal{U}(\mathcal{M})$	Utility of data using sanitization mechanism $\mathcal{M}$
$\mathcal{P}(\mathcal{M})$	Privacy attained using sanitization mechanism $\mathcal{M}$
ρ	Utility level
ϵ (or, ϵ_i)	Privacy level
$\mathbf{P}_{\mathbf{x}}$	CRLB of $\mathbf{x}$
$\mathbf{J}_{\mathbf{x}}$	FIM of $\mathbf{x}$
$\mathcal{A}$	Auxiliary information
S	Number of agents in MAS
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbf{y}_i$ (or, $\mathbf{y}_i(t)$)	Observation of i-th agent (at t-th instant)
n	Dimension of observation data tuple or trace size
m	Dimension of sanitized data tuple
$\mathcal{S}$	Set of all locations
$\mathcal{T}$	Set of all privacy mechanisms
$\mathcal{L}$	Set of all possible traces
$\mathfrak{T}$	Actual trace of user
$\mathfrak{T}'$	Actual trace estimated by adversary
$\mathcal{S}$	Set of datapoints with cardinality S

$\rho(.,.)$	Distance metric (acts as a measure of utility in reconstruction attacks)
M_p	Transition probability
G	Perturbation probability
Pr	Probability
$P_t(i)$	Residing probability
pr_t^- (pr_t^+)	Prior (posterior) probability
Δq	Global sensitivity
$N_R, N_R^{'}$	Set of nearest neighbours of the raw and reconstructed datapoint respectively
$u_i(t)$	Local uncertainty measure
$\tilde{\mathbf{y}}_i(t)$	Sanitized tuple
t^*	Total time
w_t	Timestep importance weight
$\mathcal{O}(.)$	Order of
$\hat{\mathbf{x}}$	Estimate of $\mathbf{x}$
$\Delta_i(t)$	Influence on fusion center estimate

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This chapter introduces multi-agent systems (MAS), their architecture and the privacy-utility concerns prevalent in a MAS. The main contributions of this thesis are also outlined in the chapter followed by a brief outline of the chapters of the thesis.

The growing adoption of distributed intelligent systems in multiple sectors such as smart cities [38], the Internet of Things [29], industrial automation [59], and healthcare systems [7] has accelerated the use of collaborative architectures. Multi-agent systems (MAS) have, therefore, emerged as a fundamental paradigm for large scale monitoring and decision making. In MAS, multiple autonomous agents observe local phenomena and report the information to a central hub for detailed analysis and situational awareness. Although this collaborative framework improves scalability, robustness, and estimation performance, it also causes significant privacy vulnerabilities.

From a privacy perspective, two distinct concerns arise in MAS. The first relates to the explicit disclosure of private parameters, where agents intentionally transmit certain intrinsic attributes along with their observations to improve global inference accuracy. Although such disclosure may be beneficial for system performance, it directly exposes sensitive agent-specific information. The second concern pertains to inference privacy, which arises when the transmitted data enables the extraction of additional sensitive information beyond what was intended to be revealed. Even if only task-relevant information is shared, the fusion center or an adversary may exploit statistical dependencies, structural correlations, or auxiliary side information to infer latent private attributes. In this case, privacy is compromised not through direct disclosure, but through unintended secondary inference. This unintended disclosure of private or sensitive information is referred to as an inference privacy breach. This thesis addresses the problem of inference privacy in MAS.

In this chapter, we introduce the fundamental concepts of MAS and formally define the inference privacy preservation problem. We discuss the inherent privacy-utility tradeoff problem that arises due to data sanitization mechanisms designed to address inference privacy attacks. Further, we discuss the challenges involved in designing inference privacy preserving mechanisms for MAS and summarize our contributions.

1.1 Multi-Agent System (MAS)

A MAS [19] is composed of multiple autonomous agents with limited observational and computational powers. These agent based systems are widely used for distributed inference [54]. In a typical distributed inference architecture, agents collect local observations and transmit the information to a central entity, commonly called a fusion center (or aggregator). The fusion center integrates the information received from all agents to estimate, detect, or classify one or more underlying system parameters, thus completing the inference process.

1.1.1 Healthcare monitoring system

Consider a remote healthcare monitoring system where a subject is equipped with multiple wearable sensors. Each sensor measures a specific physiological signal, such as heart rate, blood pressure, glucose level, or body temperature, acting as an agent. The underlying system parameter corresponds to the overall physiological state of the person, which can include indicators such as cardiovascular condition, metabolic stability, or stress level. To allow accurate inference, agents may also need to provide private contextual information such as activity levels, duration of sleep, dietary intake, and medication adherence. The agents transmit the information to a fusion center, such as a hospital server or a cloud-based medical platform. The fusion center, which may also have access to the person's medical history, performs joint inference to estimate the person's current health condition, detect anomalies, or predict possible medical events.

1.1.2 Smart electricity management system

Consider a smart grid system where each household is equipped with a smart meter that functions as an autonomous agent. The smart meter records local electricity consumption patterns, possibly at fine temporal granularity. The readings or processed summaries are transmitted to a central utility server that acts as the fusion center. This central unit aggregates data from all households to perform distributed inference tasks such as load forecasting, demand-response optimization, peak load management, and anomaly detection. Through this coordinated data collection and aggregation process, the system enables efficient grid management while relying on distributed observations from multiple agents.

1.1.3 Implicit privacy risk

The examples discussed earlier illustrate that privacy risks in MASs are often subtle and cumulative rather than explicit. In principle, a curious fusion center can exploit the collected data to construct detailed profiles of individual agents without their prior consent. Such attacks may leverage not only the currently reported data but also the temporal patterns and statistical correlations present in previously communicated observation tuples.

In the healthcare setting, physiological measurements and contextual attributes, when observed over time, form a high-dimensional behavioral signal. An adversary can reconstruct the individual's lifestyle habits, travel frequency, and other socioeconomic indicators, thus exposing sensitive personal information beyond the intended clinical objective. In the smart grid setting, similar statis-

tical exploitation can reveal occupancy behavior, appliance usage signatures, or socioeconomic characteristics of a household.

These inferences arise from structural regularities in the data, rather than from any direct disclosure of private attributes. The privacy loss in such systems is fundamentally inferential in nature. This class of attacks is referred to as inference privacy breaches. To better position the problem addressed in this thesis, we will now outline the differences between data privacy and inference privacy; followed by the distinction between inference privacy and other relevant notions, including security and confidentiality. This will consequently justify why the the preservation of inference privacy requires a different approach.

1.1.3.1 Data privacy vs. inference privacy

The primary focus of data privacy is to prevent unauthorized individuals from accessing the data. On the other hand, preservation of inference privacy requires the prevention of illegitimate inference of secondary information from legitimately shared data. We further illustrate the difference with an example.

In case of the electricity management system, houses from several different localities send their hourly power consumption data to a central base station. An unauthorized individual accessing the data falls under a data privacy breach. However, the base station can also create a profile of an individual house by exploiting the consumption information to infer about their daily routine and possible financial status. Such a violation falls into the category of inference privacy breach.

The set of adversaries in case of inference privacy is thus larger than the set of possible adversaries for data privacy. Since authorized individuals can also carry out inference privacy breaches, there is no trusted party (TP). This implies that the set of possible adversaries in case of inference privacy breach, includes everyone other than the data owner.

1.1.3.2 Privacy objectives and security properties

Inference privacy, security and confidentiality represent different objectives. Security is achieved when the system is protected against unauthorized access, modification and disruption. Confidentiality ensures that only authorized entities can access the transmitted data, and is commonly achieved using cryptographic techniques. On the other hand, inference privacy focuses on limiting the information that can be inferred about an individual even when the data is legitimately shared. Even by ensuring confidentiality through encryption, an authorized fusion center may infer secondary sensitive information after decryption. Therefore, preserving confidentiality alone does not guarantee inference privacy.

Reconstruction resistance is an empirical privacy property that reflects the difficulty faced by an adversary in reconstructing the original observations from the released data. Since accurate reconstruction of the raw observations facilitates inference of sensitive information, higher reconstruction resistance corresponds to a lower risk of inference privacy breaches. Throughout this thesis, reconstruction based metrics are therefore employed as empirical privacy indicators.

In contrast to inference privacy, Byzantine failures arise when one or more agents behave arbitrarily or maliciously by transmitting inconsistent information with the objective of disrupting the inference process. In this thesis, participating agents are assumed to execute the sanitization mechanism correctly. The fusion center is also assumed to faithfully execute the intended inference task using the sanitized observations, while remaining curious with respect to inferring additional sensitive information.

1.1.3.3 Threat model

The literature considers several adversarial models. A semi-honest adversary is one who corrupts parties but follows the protocol as specified. The corrupt parties run the protocol honestly but they may try to learn as much as possible from the messages they receive from other parties. A colluding adversary consists of multiple entities that cooperate by sharing their observations and auxiliary information to improve the effectiveness of inference attacks.

In contrast, this thesis considers an honest-but-curious adversary. Such an adversary may represent any entity other than the data-owner, including a fusion center, service provider, or external observer. The mechanisms developed in this thesis are analyzed under the presence of adversaries that satisfy the following.

1. The adversary knows the sanitization mechanism $\mathcal{M}$.
2. The adversary has access only to the sanitized reports $\{\tilde{\mathbf{y}}_i(t)\}$ transmitted by the agents and may exploit this observation history.
3. The adversary may possess auxiliary information like statistical characteristics of the monitored process, temporal correlations, spatial correlations, and application domain knowledge.
4. The adversary cannot directly access any raw information before sanitization. The privacy preserving mechanism is therefore assumed to execute correctly at every agent. This also implies that attacks involving known plaintext-ciphertext pairs cannot be executed by the adversary.

1.2 Inference privacy

Let $\mathbf{y}_i(t)$ denote the local observation generated by agent i at the t-th step.

$$\mathbf{y}_i(t) = (p_{1t}, p_{2t}, \ldots, p_{kt}, g_{1t}, \ldots, g_{(n-k)t}) \in \mathbb{R}^n.$$

Here, $\{p_{jt}\}_{j=1}^{k}$ denote the private parameters associated with the agent a_i, and $\{g_{jt}\}_{j=1}^{n-k}$ represent the observations of the agent. Let $\mathbf{x}$ denote the global system parameter to be inferred by the fusion center and p_i denote a sensitive (private) variable associated with agent a_i. To protect the inference of secondary information of each agent, an inference privacy preserving mechanism needs to be implemented. The agent generates a sanitized report $\tilde{\mathbf{y}}_i$ using a privacy mechanism $\mathcal{M}$.

Definition 1.1 *(**Inference privacy breach**) An inference privacy breach occurs if the observation $\tilde{\mathbf{y}}_i$ enables an adversary to significantly update its belief about the private variable p_i. Formally, an inference privacy breach occurs if*

$$Pr(p_i|\tilde{\mathbf{y}}_i) \neq Pr(p_i),$$

or equivalently, if

$$MI(p_i; \tilde{\mathbf{y}}_i) > 0,$$

where $MI(.;.)$ denotes mutual information.

Depending on the application, p_i may represent the exact location of a mobile agent, a mobility trajectory, data streams, private attributes of the agent, or any other sensitive variable linked with the agent. Inference privacy concerns the unintended leakage of information about a sensitive variable p_i through shared data $\tilde{\mathbf{y}}_i$ that is primarily intended to estimate the global parameter $\mathbf{x}$.

Let $\mathcal{M}$ be a sanitization function, the modified tuple of the i-th agent takes the following form:

$$\tilde{\mathbf{y}}_i = \mathcal{M}(\mathbf{y}_i) = (p'_{1t}, p'_{2t}, \ldots, p'_{k't}, g'_{1t}, \ldots, g'_{(m-k')t}) \in \mathbb{R}^m.$$

Here, $\{p'_{jt}\}_{j=1}^{k'}$ and $\{g'_{jt}\}_{j=1}^{m-k'}$ denote the sanitized components corresponding to the private and task-relevant attributes, respectively. The dimension m of the modified tuple depends on the type of sanitization function used. In particular, $\mathcal{M}$ may compress or perturb the original observation vector y_i, potentially altering both its structure and dimensionality. A basic MAS architecture is outlined in Figure 1.1.

Sanitization inevitably alters the original data distribution, reducing the accuracy of global inference. This modification introduces a fundamental trade-off between privacy preservation and utility. In the next section, we examine this privacy-utility tradeoff in detail and formalize the associated optimization problem.

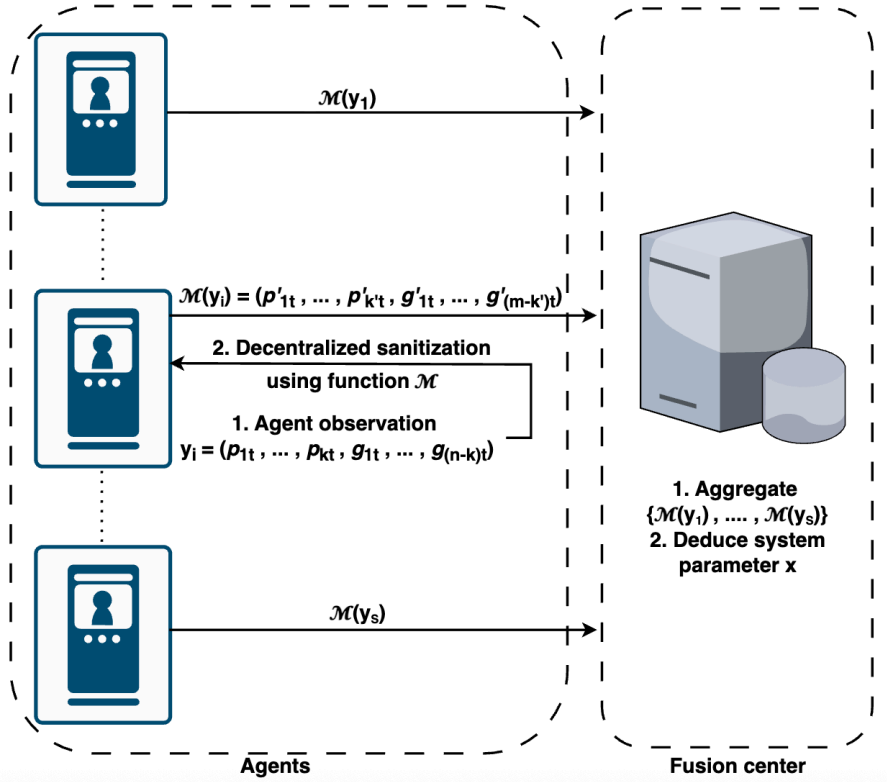


Figure 1.1: MAS Architecture.

1.3 Privacy-utility tradeoff

Privacy preserving mechanisms inevitably modify the raw data to prevent leakage of secondary information. This modification alters the statistical properties of the data and may degrade the accuracy of global inference. Therefore, while satisfying privacy requirements, it is equally important to preserve utility. That is, the fusion center should still be able to reliably deduce the system parameters using the sanitized data.

Let $\mathcal{U}(\mathcal{M})$ and $\mathcal{P}(\mathcal{M})$ be the utility and privacy values attained using the sanitization mechanism $\mathcal{M}$. The two quantities, utility and privacy, exhibit a competing relationship: increasing privacy protection typically reduces inference accuracy, and vice versa. The tradeoff between utility and privacy can be formulated as an optimization problem. The choice of optimization formulation depends on the system design objective and the relative emphasis placed on privacy and utility.

In utility critical applications, privacy may be treated as a strict constraint while maximizing utility.

$$\begin{aligned} \max_{\mathcal{M}\in\mathcal{T}} \quad & \mathcal{U}(\mathcal{M}), \\ \text{s.t.} \quad & \mathcal{P}(\mathcal{M}) \geq \epsilon. \end{aligned}$$

Conversely, in privacy-sensitive systems, privacy is maximized subject to a minimum acceptable inference accuracy. Thus, the structure of the optimization problem reflects the operational priorities of the application domain.

$$\begin{aligned} \max_{\mathcal{M}\in\mathcal{T}} \quad & \mathcal{P}(\mathcal{M}), \\ \text{s.t.} \quad & \mathcal{U}(\mathcal{M}) \geq \rho. \end{aligned}$$

To solve the optimization problem, both utility and privacy must be defined precisely in mathematical terms. In the literature [70, 20, 71], this quantification has been done using advanced statistical functions, such as the Fisher information matrix (FIM) and Cramér-Rao lower bounds (CRLB) [69].

Definition 1.2 *(**CRLB-based utility and privacy**) Let* $\mathbf{P}_{\mathbf{x}}(\mathcal{M})$ *denote the CRLB matrix for estimating* $\mathbf{x}$ *from the collection of sanitized observations* $\{\tilde{\mathbf{y}}_i\}$*, and let* $\mathbf{P}_{\mathbf{x}}^{(0)}$ *denote the corresponding CRLB before sanitization. Similarly, let* $\mathbf{P}_{\mathbf{z}_i}^{(0)}$ *and* $\mathbf{P}_{\mathbf{z}_i}(\mathcal{M})$ *denote the corresponding CRLB matrices before and after sanitization. The* utility *of the mechanism* $\mathcal{M}$ *is defined as*

$$\mathcal{U}(\mathcal{M}) = \Phi_u\big(\mathbf{P}_{\mathbf{x}}^{(0)}, \mathbf{P}_{\mathbf{x}}(\mathcal{M})\big)\,,$$

where Φ_u *is a scalar functional that is non-increasing in* $\mathbf{P}_{\mathbf{x}}(\mathcal{M})$ *with respect to the Loewner order.*

The privacy *under mechanism* $\mathcal{M}$ *is defined as*

$$\mathcal{P}(\mathcal{M}) = \Psi_p\big(\mathbf{P}_{\mathbf{z}_i}^{(0)}, \mathbf{P}_{\mathbf{z}_i}(\mathcal{M})\big)\,,$$

where Ψ_p *is a scalar functional that is non-decreasing in* $\mathbf{P}_{\mathbf{z}_i}(\mathcal{M})$ *with respect to the Loewner order. CRLB is obtained as the inverse of the corresponding FIM, provided that the FIM is nonsingular.*

Utility increases with Fisher information of $\mathbf{x}$, while privacy increases with the Cramér-Rao lower bound of p_i. Although these metrics provide strong theoretical guarantees, they are computationally demanding and may not be suitable for resource constrained agents. This limitation highlights the need for alternative

quantification methods that are computationally efficient but retain comparable theoretical rigor.

1.4 Sanitization function design requirements

The preservation of inference privacy requires that the sanitization mechanism satisfies the structural, statistical, and computational constraints imposed by the system architecture.

1. Decentralized architecture: Due to the nature of an inference privacy breach, relying on a centralized privacy preserving architecture is impractical. Moreover, a centralized framework also introduces a single point of failure and leads to a risk of large scale privacy compromise. Thus, privacy mechanism $\mathcal{M}$ in a MAS must be locally enforced at the agent level. That is, agent i uses a copy of $\mathcal{M}$, namely, $\mathcal{M}_i$, to generate

$$\tilde{\mathbf{y}}_i = \mathcal{M}_i(\mathbf{y}_i),$$

with no dependency on the data from agent j.

2. Privacy under data correlation: Agent observations $\mathbf{y}_i(t)_{t\geq 1}$ at time t and $t+1$, are generally temporally and structurally correlated. Let

$$\tilde{\mathbf{y}}_i(1:t^*) = \{\mathcal{M}(\mathbf{y}_i(1)), \ldots, \mathcal{M}(\mathbf{y}_i(t^*))\}$$

denote the sequence of sanitized reports. The mechanism must ensure that privacy guarantees do not degrade over time. That is, the total information leakage about the sensitive variable p_i remains uniformly bounded over time, even when multiple sanitized reports are accumulated,

$$MI(p_i; \tilde{\mathbf{y}}_i(1:t^*)) \leq \zeta \quad \forall t^*.$$

Inter-agent correlations may lead to reduced privacy preservation as well. That is, an adversary may infer about one agent from the data received from others. In this thesis, however, the agents are assumed to operate independently. Consequently, the proposed mechanisms consider only the spatio-temporal correlations present within the data streams of each individual agent.

3. Computationally efficient: Agents in a distributed system are constrained by their limited resources, including limited computational power, restricted

memory capacity, and energy availability. Unlike the central fusion center, the agents cannot support computationally intensive or storage heavy data processing frameworks.

Excessive computational overhead increases latency and leads to rapid energy depletion, thereby affecting the reliability of the system. Since the mechanism needs to be decentralized, it must be lightweight and compatible with low powered devices.

4. Stream based processing: In most MAS applications, data observation is sequential, with the data being time sensitive. Dependency on data history introduces latency, degrades utility and exhausts the memory capacity of the agents and leads to a failure of the system. The mechanism should operate in an online manner, processing each data tuple individually with minimal dependency on historic data.

   That is, $\mathcal{M}$ should operate as

   $$\tilde{\mathbf{y}}_i(t) = \mathcal{M}(\mathbf{y}_i(t)),$$

   without requiring access to $\{\mathbf{y}_i(\tau)\}_{\tau=1}^{t-1}$.

   Formally, $\mathcal{M}$ should not require require batch processing over a dataset $\mathcal{D}_i = \{\mathbf{y}_i(1), \mathbf{y}_i(2), \ldots, \mathbf{y}_i(t^*)\}$. This will eliminate the temporary storage of the raw data, which reduces substantial storage buffers.

5. Scalable: The mechanism must remain effective with the growth in the number of agents in the MAS. The computational and communication overhead should grow at most proportionally with the system size. That is, if S is the total number of agents, the mechanism should not introduce pairwise dependencies of the form $\mathcal{O}(S^2)$. This ensures applicability in large-scale, dense and dynamic MASs.

6. Robust to reconstruction attacks: The privacy mechanism should prevent adversaries from reconstructing sensitive information from the sanitized tuple. Robustness requires that the privacy leakage is bounded by a small $\zeta > 0$. That is, it must provide strong resistance against reconstruction attacks even when the adversary exploits data correlations and other auxiliary information ($\mathcal{A}$) over time. This implies that $\mathcal{M}$ should satisfy

   $$MI(p_i; \tilde{\mathbf{y}}_i, \mathcal{A}) \leq \zeta.$$

   The mechanism should also prevent the deduction of secondary information to ensure efficiency in case of continued deployment over a period of time.

7. Provide a principled privacy-utility tradeoff: The mechanism should provide privacy guarantees while ensuring estimation accuracy at the fusion center. Excessive perturbation of the raw data severely degrades the data utility, while insufficient privacy levels lead to enhanced risk of breaches.

   Equivalently, the method should be an optimum solution for

   $$\min_{\mathcal{M}} \mathcal{U} + \lambda \mathcal{P},$$

   where λ quantifies the permissible tradeoff. The mechanism should have a tunable parameter for utility or privacy or it should inherently provide an optimal tradeoff through a principled optimization based framework.

Having outlined the fundamental requirements for a privacy mechanism to be deployable in a MAS, we now briefly highlight the shortcomings of the existing methods. These shortcomings motivate the need for new privacy preserving mechanisms.

1.5 Challenges in applying existing methods

The existing privacy preserving mechanisms pose notable limitations when directly applied to a MAS setting. Many were originally proposed for centralized or static systems, and do not align with the decentralized, resource constrained, streaming setting that is encountered in a MAS. We briefly highlight the key drawbacks below.

1. Differential privacy (DP) method for mobile agents: In continuous query settings where mobile agents transmit their data at regular intervals for a location based service (LBS), the data exhibits strong spatio-temporal correlation. An adversary with access to historical data can deduce the mobility profile of the agent. Repeated application of a standard DP method with a uniformly allocated per timestep privacy budget results in cumulative privacy loss due to sequential composition across timesteps. Consequently, the combined effect of budget composition and inherent spatio-temporal dependence leads to progressively weakened inference privacy guarantees.

   That is, DP fails to uphold Requirement 2 in case of repeated application to continuous queries which are spatio-temporally correlated. Repeated application of a standard DP mechanism $\mathcal{M}$ yields

   $$MI(p_i; \tilde{\mathbf{y}}_i(1:t^*)) > \zeta \quad \text{for some } t^*.$$

2. Uniform DP budget allocation: Most DP based methods allocate privacy budgets uniformly across timesteps. Uniform allocation results in progressive privacy loss when composed sequentially over time. A uniform allocation constrained by a fixed cumulative privacy budget does not differentiate between informative and non informative observations. This leads to inefficient noise injection. The relative contribution of different agents to global inference is ignored, thereby failing to provide a principled balance of privacy and utility.

   An uniform budget allocation fails to fulfill Requirement 7. In case of distributed inference, it must be ensured that the deduction of the system parameter is possible using the sanitized data. Uniform allocation of privacy budget fails to provide that guarantee.

3. Dimension reduction: Many existing privacy preserving mechanisms rely on compression. By compression, we refer to the reduction of the dimension of the data tuple prior to transmission. A majority of such methods require a significant portion of the dataset to be available prior to its application.

   That is, contrary to Requirement 4, such methods require access to $\{\mathbf{y}_i(\tau)\}_{\tau=1}^{t-1}$. These methods require batch processing over a dataset $\mathcal{D}_i = \{\mathbf{y}_i(1), \mathbf{y}_i(2), \ldots, \mathbf{y}_i(t^*)\}$.

   On the other hand, compression methods which generate a representative of the data tuple, require significant computational overhead, thereby failing Requirement 3. Both of these requirements pose serious challenges in resource constrained temporal data streaming settings like MAS. On the other hand, hashing based compression algorithms lack tunable privacy-utility parameters, making controlled tradeoff design infeasible, violating Requirement 7.

4. Cryptographic methods: While key based symmetric and asymmetric encryption schemes ensure data confidentiality, they fail to enforce inference privacy. Decrypted outputs which are required for system parameter inference and the correlated ciphertext patterns can be exploited to carry out inference privacy breaches. That is, key based encryption techniques that require data decryption for processing fail to meet Requirement 6. Formally, if $\mathcal{M}$ is such a method,

$$MI(p_i; \mathbf{y}_i, \mathcal{A}) \not\leq \zeta.$$

   Although homomorphic encryption allows computation on the encrypted data, it incurs significant computational overhead, thereby failing Requirement 3. These drawbacks make them unsuitable for decentralized and re-

source constrained MASs. Similarly, lattice based constructions and complex cryptographic hash functions lack tunable privacy–utility parameters and introduce scalability and efficiency challenges for lightweight agents.

5. Federated learning (FL): FL enables distributed training by sharing model updates instead of raw data. However, it remains vulnerable to privacy leakage through gradients and update vectors, which can be exploited to carry out reconstruction attacks and thus violate Requirement 6. DP based methods such as differentially private stochastic gradient descent (DP-SGD) mitigate the leakage by clipping gradients and adding noise. Fixed or improperly tuned parameters often lead to suboptimal privacy–utility tradeoffs. Adaptive DP-FL methods improve convergence by dynamically adjusting clipping norms or noise levels, but they lack real time privacy risk control and are primarily designed for model training rather than streaming MAS data. Another major drawback is the significant computational and communication overhead, contrary to Requirement 3 and thus limiting their suitability for lightweight, resource constrained MASs.

This discussion highlights that the direct application of the existing methods to MAS introduces practical challenges and limitations. They have limited applicability in decentralized, resource constrained systems that stream correlated data. These shortcomings underscore the need for a privacy mechanism tailored specifically for distributed multi-agent environments.

1.6 Contributions

Our main contributions in this thesis can be summarized as follows.

1. We illustrate theoretically that standard differential privacy methods fail in systems involving a continuous stream of location data from mobile agents. This is due to sequential composition of the fixed per timestep privacy budget and spatio-temporal correlation. We propose a perturbation generation mechanism for ensuring privacy for location data that exploits the mobility profile and the residing probability of the agent. The proposed method resorts to DP if and only if when an appropriate candidate output is not available. We further show empirically that it performs better than standard DP method.

2. We propose a per timestep adaptive differential privacy budget allocation method based on the informativeness of the data shared by the agent. This method prioritizes data utility while ensuring that despite sequential composition, the cumulative privacy requirement is upheld.

3. We introduce a dynamic projection method. This method uses a new projection matrix for each sanitization step for matrix multiplication based projection. However, matrix regeneration is replaced by in-place updation of the projection matrix after each sanitization step. The updation is done using the current raw and sanitized tuple following which, the new updated matrix is used for sanitization at the consecutive timestep.

4. We propose an alternative, computationally efficient, geometric measure of data distortion based on cosine similarity to quantify utility and privacy. Following this, we incorporate the cosine based privacy requirement to propose an adaptive modulo hash function which eliminates the selection of a fixed divisor. It instead calculates the hash value by solving a quadratic equation. The quadratic equation is formed by incorporating the privacy requirement and the tuple parameters.

Design Requirement	**Chapter**	**Reasoning**
Decentralized architecture	Chapters 3, 5, and 6	The sanitization step is completely performed locally before transmission to the fusion center.
Privacy under data correlation	Chapter 3, 4, and 6	Privacy preserved for continuous data release while accounting for spatio/temporal correlations.
Computationally efficient	Chapter 6	Lightweight sanitization suitable for resource-constrained agents.
Stream-based processing	Chapters 3 - 6	Each observation is sanitized independently without batch processing or storage of historical data.
Scalable	Chapters 4 - 6	Computational and communication costs grow linearly with the number of agents.
Robust to reconstruction attacks	Chapters 3 - 5	Reduced reconstruction capability compared with baseline mechanisms. Empirically shown using metrics (breach count, displacement and resemblance).
Principled privacy-utility tradeoff	Chapters 3 - 6	Demonstrates favourable privacy-utility tradeoff through theoretical analysis and/or experimental evaluation.

Table 1.1: Traceability of the design requirements with the proposed mechanisms.

1.7 Thesis Organization

This thesis is organized into nine chapters. **Chapter 1** presents a general introduction about the need for inference privacy preservation. It also discusses the motivation and objectives of the methods outlined in the thesis. The remaining chapters of this thesis are organized as follows.

In **Chapter 2** of the thesis, we discuss the available literature pertaining to the topic. We look at the different privacy preserving mechanisms available along with their limitations. We further motivate our problem and the need of our work in today's distributed inference scenario.

In **Chapter 3** we study the applicability of standard differential privacy based methods in systems where mobile agents continuously stream their location data to a central fusion center. We theoretically show that the standard differential privacy based mechanisms with fixed per timestep budget allocation, fail to guarantee optimal level of privacy due to sequential composition of privacy budgets and spatio-temporal correlations. We conclude the chapter by proposing and empirically proving the efficacy of a perturbation generation method for such mobile systems. The proposed perturbation exploits the mobility profile of the user, and falls back to DP if and only if a feasible candidate output is not found.

In **Chapter 4** of the thesis, we question whether a fixed per timestep privacy budget allocation in case of standard DP can be modified to an adaptive budget allocation method that would prioritize the signal informativeness. This chapter proposes such an algorithm which retains the fixed agent privacy requirement under sequential composition while allocating the budget in a manner that incorporates the signal utility. This chapter concludes by empirically showing that an adaptive budget allocation fares better than representative baselines.

Following the differential privacy based methods, in **Chapter 5** of the thesis, we propose a learning based dynamic projection method (DPM). We first introduce the robust concept problem, and highlight the analogy between this problem and the MAS model. Following this, we describe the neuronal random projection (NRP) method, wherein the projection matrix used for dimension reduction is regenerated for each sanitization step. In DPM, we eliminate the need for regeneration by updating the matrix from the previous step using the raw and sanitized data of the current step. The updation rule makes this method data dependent.

In **Chapter 6** of the thesis, we develop a hashing based algorithm for dimension reduction and subsequent preservation of privacy. This method is data dependent, and a novel spin-off of the modulo based hashing algorithm. Termed as adaptive modulo hash function, this method does not fix a divisor for hash calculation. It solves a quadratic equation formed by using elements of the data

tuple and the privacy requirement of the agent. Being computationally cheaper than the projection methods, this method is well suited for MASs.

The main findings of this thesis are summarized in **Chapter 7**, where the key contributions to the area of inference privacy preservation are discussed. This chapter outlines potential avenues for future research, including ways in which the concepts introduced here can be further developed to advance the field. The pertinent references supporting this work are listed in the **Bibliography**.

Chapter 2

Related Work

This chapter reviews the existing literature on privacy preservation in MAS and their drawbacks. In particular, differential privacy based method, dimension reduction mechanisms, cryptographic methods have been predominantly covered. The utility-privacy tradeoff problem has also been discussed.

Privacy preservation in MAS has received significant attention owing to their growing popularity in data sensitive fields including healthcare [7] and finance [2]. A privacy preserving mechanism designed for MAS must satisfy several critical requirements: it should operate in a decentralized manner, impose minimal computational overhead on resource constrained agents, scale efficiently with the number of agents, and remain robust against reconstruction based inference attacks. Furthermore, such mechanisms must be resilient to adversaries exploiting the spatio-temporal correlations in agent data. The types of possible correlations have been explored by Zhang et al. [79].

A real world MAS scenario where we observe strong spatio-temporal correlation is in wireless sensor networks (WSNs) [12]. The continuous evolution of sensor networks has led to their widespread use for data collection. Sensors transmit packets to report events [37], however, the contextual information like the time of packet creation and the location of the sensor node raise privacy concerns. Correlation between successive packets along with the spatio-temporal evolution of the traffic can be exploited by an adversary. Such a continuous query scenario is prevalent in all MAS models where the agents report data at regular intervals.

Preservation of inference privacy in distributed settings necessitates the enforcement of a decentralized sanitization mechanism. However, any modification applied to the data must also preserve its utility, ensuring that the underlying system objectives and inference tasks remain achievable.

Popular existing mechanisms for the preservation of privacy include differential privacy (DP) with uniform per timestep budget allocation, dimension reduction methods and cryptographic protocols. However, when applied to distributed settings like MAS, which handle a substantial amount of correlated data, the existing methods exhibit critical flaws.

The existing privacy preserving mechanisms, in their standard form, do not align with the decentralized, resource constrained, and streaming nature of MAS environments. In continuous query scenarios with strong spatio-temporal correlation, repeated application of standard DP mechanisms results in cumulative privacy loss, while uniform privacy budget allocation fails to account for temporal signal variations and agent heterogeneity, leading to suboptimal utility–privacy tradeoffs. Compression based methods either assume partial dataset access or introduce computational overhead incompatible with lightweight agents. Cryptographic approaches ensure confidentiality but do not prevent inference from released outputs and often are computationally expensive. Similarly, federated learning frameworks remain vulnerable to reconstruction attacks and impose significant communication and processing costs. These limitations motivate the need for decentralized, computationally efficient, and correlation aware privacy mecha-

nisms tailored specifically for streaming MAS settings.

This chapter is organized according to the major class of the privacy-preserving mechanisms, namely DP-based methods, cryptographic, reconstruction resistance via dimension reduction, and learning based methods as these directly motivate the techniques proposed in the subsequent chapters. These categories can be further subdivided into two complementary perspectives. The first comprises algorithms providing formal privacy guarantees. Examples include DP, local differential privacy (LDP), Rényi differential privacy (RDP), zero-concentrated differential privacy (zCDP), mutual-information privacy (MIP) which bounds the information leakage between sensitive data and released data and cryptographic techniques such as homomorphic encryption and secure multi-party computation. These methods provide guarantees under clearly specified assumptions regarding the adversary and system model. The second category consists of algorithms whose efficacy is measured using empirical privacy indicators like reconstruction resistance and inference accuracy. These approaches generally demonstrate privacy empirically against specific adversaries but do not necessarily provide worst-case theoretical guarantees.

2.1 Differential privacy (DP)

The use of DP [22] for privacy preservation has gained extensive popularity in the last decade [15, 76]. It was initially introduced for statistical databases. Rather than constituting a specific mechanism, DP represents a formal and quantifiable guarantee of privacy under carefully designed data release procedures. DP guarantees that the inclusion of an individual data tuple in a database does not lead to an increased risk of privacy breach [18].

In DP terminology, a dataset is a collection of data points pertaining to a person, entity, or event. Applying a DP based mechanism requires the identification of neighbouring datasets [24]. In classical DP, two datasets are said to be neighbours if one dataset is a subset of the other and the larger dataset contains exactly one additional entry. However, in most real life MAS scenarios, the identification of such neighbouring datasets is not feasible.

Standard differential privacy assumes that a trusted data curator perturbs the collected dataset before publication, whereas local differential privacy (LDP) performs perturbation at the data source before transmission. Whereas traditional DP works on whole or a part of the dataset, LDP does so in decentralized environments by enabling agents to perturb their data independently before communication [21].

Most of the DP techniques proposed in the literature are aimed at point queries.

This means the queries are assumed to be independent of each other, that is, there is no correlation between subsequent queries. For such queries, DP can provide a provable privacy level. A classical way of enforcing DP is by the addition of calibrated noise generated randomly from a suitable probability distribution.

We will now look at the DP based mechanisms proposed for the real life MAS scenarios and their drawbacks.

2.1.1 For continuous queries from mobile agents

The term continuous queries has different interpretations in the literature. By continuous, one may refer to queries whose results are continuous [27]. However, by continuous queries we refer to scenarios where successive data generated by a node have a spatio-temporal correlation.

Mobile agents rarely use single point queries. For example, a moving user in the city may perform different activities such as having lunch, shopping, and visiting friends. During these activities, the user performs several queries, such as searching for restaurants, malls, and getting driving directions.

For streaming location data, the notion of adjacency also becomes important. Event-level adjacency considers two datasets adjacent if they differ in a single location record. This, therefore, protects an individual location update while leaving the remainder of the trajectory unchanged. Trajectory-level privacy considers one complete trajectory, providing privacy protection for an entire movement path or trip. That is, one entire trajectory is an unit to be protected. User-level adjacency is stronger, treating two datasets as adjacent when they differ in all the records associated with one user, thereby protecting the user's complete collection of trajectories. Consequently, the appropriate notion of adjacency depends on whether the goal is to protect individual location updates, complete trajectories, or all data contributed by a user.

One may use a noise addition based privacy mechanism for each such location independently, thereby providing event-level privacy. However, the level of privacy attained will eventually degrade due to the correlation of the locations. Andrés et al. [5] introduce the concept of geo-indistinguishability to quantify the loss when a privacy mechanism is applied repeatedly. Geo-indistinguishability adapts DP to geographic coordinates. Geo-indistinguishability requires that if two locations are geographically close, they are more indistinguishable. The authors state that if a privacy mechanism adds ϵ geo-indistinguishable noise when applied independently, then the privacy loss when applied to n' locations is $n'\epsilon$ geo-indistinguishable. However, the authors do not give a formal proof.

Chatzikokolakis et al. [13] further exploit the concept of geo-indistinguishability and propose the use of correlation between locations to their

advantage. The authors propose to use the mobility profile and previous locations of the user to generate the perturbed location instead of generating it independently. Their proposed predictive method contains three components: a predictive function, a test mechanism and a noise generating mechanism. There is a preset budget for testing with no provision to reset the budget after it is exhausted. This implies that the user needs to stop using the system once the testing budget is exhausted.

Xiao et al. [75] introduce the use of δ-location set and sensitivity hull to handle the issue of neighbouring datasets for location data. The δ-location set includes the most probable locations. The authors use the Markov model to handle the temporal correlation between user data which is a hidden Markov model since the original user location is unknown to the adversary. Their proposed approach considers the mobility profile, but the spatial constraints are overlooked. For example, if a user is moving on a street, then an attacker can exclude the locations that are not accessible from the road.

In industrial WSN [36], the nodes are generally mobile such as workers and mobile devices; and the process of data generation involves navigating sensitive location information. Wang et al. [73] proposed to split the geospatial data domain of WSNs into cells and perturb the data of each cell by random noise following a Laplace distribution. Chakraborty et al. [12] propose a temporal privacy preserving mechanism for WSN data by delaying the packets en route to every node. The privacy mechanisms of WSN data available in the literature handle spatial and temporal privacy independently but do not address their correlation.

An effective temporal privacy mechanism should change the traffic pattern to reduce the correlation between the creation time and reception time of a packet. One possible way to reduce the temporal correlation is by introducing random delays using DP [12] while forwarding the packets. From a utility viewpoint, introducing delays will make the service unsuitable for real time applications. This implies that the utility of the data will be lost.

Privacy accounting determines how cumulative privacy loss is quantified when a mechanism is applied repeatedly. The sequential composition adopted throughout this thesis, provides a straightforward accumulation of privacy budgets across multiple releases. To obtain tighter cumulative privacy bounds, advanced composition theorems, privacy filters, privacy odometers, RDP, and zCDP have been proposed. These approaches differ primarily in the manner in which privacy loss is accounted for and the tightness of the resulting privacy guarantees. Their applicability depends on the underlying application, privacy requirements, and accounting framework adopted. This thesis adopts the sequential composition framework, as it provides a simple and transparent accounting of cumulative privacy loss.

Both DP and LDP often yield substantial performance degradation in distributed inference: noise accumulates over multiple communication steps, privacy budget depletion, and deteriorating estimation accuracy [16, 31]. In case of a DP based noise addition method, the probability distribution is calibrated using the privacy budget. In case of continuous data transmission from the agent to the fusion center, it should be ensured that the cumulative privacy budget is within the stipulated privacy requirement. Thus, the total privacy budget needs to be appropriately allocated between the timesteps. We will now explore how DP budget allocation affects the effectiveness of a DP mechanism in the case of streaming data.

2.1.2 Privacy budget allocation for streaming data

Traditional DP based noise addition mechanisms often rely on fixed noise levels. That is, the privacy budget allocated for each step is the same throughout the lifetime of the agent. A main concern in a MAS is the preservation of utility while maintaining a level of privacy over the communication lifetime of each agent with the fusion center.

Recent studies have investigated utility retention using DP in a decentralized framework. Jose and Simeone [43] analyze how DP perturbations affect estimation accuracy from an information theoretic perspective. Papachristou et al. [57] propose a DP mechanism in which each agent perturbs locally computed sufficient statistics, establishing formal privacy risk tradeoffs. The utility of such perturbed data is fundamentally limited by the amount of noise injected, particularly when privacy budgets are strictly enforced.

An uniform budget per communication step degrades utility in case of agents dealing with heterogeneous or temporally evolving datasets [74, 56]. Such data is frequently observed in real life MAS scenarios, and handling them to maximize utility while preserving privacy levels is crucial to the functionality of the MAS.

Temporal correlation in the data introduces additional challenges in the design of the privacy mechanism. Huang et al. [42] combine LDP with a shuffling mechanism to amplify privacy and reduce noise accumulation. While effective in protecting individual measurements, this approach primarily uses static noise allocation and does not exploit temporal variations in signal uncertainty, limiting utility in streaming MAS. When aggregated across multiple agents, the cumulative noise significantly distorts the estimated system parameters, reducing the overall analytical precision. To mitigate this, the adaptive noise injection (ANI) method [45] was proposed for privacy-preserving remote inference. ANI introduces input specific stochastic noise to obscure sensitive features at the client side. However, since its noise generator is trained in a data driven manner with-

out formal privacy guarantees, ANI is vulnerable to input reconstruction attacks, where adversaries may learn to approximate the original data from the perturbed representations.

The attack-aware noise injection (AANI) method [47] builds upon ANI by adaptively scaling the injected noise according to the anticipated sensitivity of potential attacks. This design improves utility by optimizing the noise magnitude for a given threat level. Nonetheless, AANI depends on predefined attack thresholds, and the absence of universal rules for selecting these thresholds limits its applicability in fully decentralized MAS environments.

Inference and reconstruction attacks further highlight the need for careful noise management [42]. In streaming MAS, static or poorly tuned noise can leave agents vulnerable, as adversaries can leverage temporal correlations across reports. A comparative summary of the approaches, highlighting their core ideas, strengths, and limitations in MAS settings, is presented in Table 2.1.

Method	**Core Idea**	**Strength**	**Limitations in MAS**
Classical DP [22, 24]	Noise calibrated by privacy budget ϵ	Formal privacy guarantee	Requires neighbouring datasets; assumes independent queries; cumulative privacy loss due to sequential composition
LDP [21]	Local perturbation before transmission	Fully decentralized privacy enforcement	Significant utility degradation in distributed inference
Geo-indistinguishability [5]	Location-dependent noise; privacy loss scales as $n'\epsilon$	Captures spatial indistinguishability	Linear privacy degradation under repeated use
Predictive DP [13]	Mobility-profile-based perturbation	Incorporates temporal correlation	Fixed testing budget; requires prior mobility model

Method	Core Idea	Strength	Limitations in MAS
δ-Location Set [75]	Sensitivity hull and Markov modeling	Handles temporal dependence formally	Ignores spatial constraints; model-dependent
WSN Spatial Perturbation [73]	Cell-based Laplace noise	Simple spatial privacy	No temporal correlation handling
WSN Temporal Privacy [12]	Random packet transmission delays	Reduces temporal leakage	Degrades real time utility
Fixed Budget Allocation [16]	Uniform allocation $\epsilon_t = \epsilon/t^*$	Easy implementation	Budget depletion; ignores data heterogeneity
Adaptive DP Allocation [74]	Dynamic noise or clipping adjustment	Improved utility	No real time privacy leakage monitoring
Shuffled LDP [42]	LDP combined with shuffling	Privacy amplification	Static noise allocation; limited temporal adaptivity
Adaptive Noise Injection (ANI) [45]	Data-driven stochastic noise	Utility-aware perturbation	No formal privacy guarantee; vulnerable to reconstruction
Attack-aware Noise Injection (AANI) [47]	Noise scaled by anticipated attack sensitivity	Improved utility under threat assumptions	Requires predefined thresholds; limited decentralization

Table 2.1: DP based mechanisms and their limitations in MAS.

2.2 Cryptographic methods

Beyond stochastic perturbation, cryptographic and coding theoretic approaches have been proposed in the literature. Traditional key based symmetric and asymmetric cryptographic methods [63] cannot be used to preserve inference privacy since they reveal the actual data completely upon decryption.

Paillier based secure aggregation [10] offers perfect data confidentiality but not inference privacy and requires expensive modular exponentiations and key management. Several studies have explored encryption based privacy preserving computation. Zhang et al. [78] integrated encryption, decryption, and neural networks for secure inference. However, the decryption required for label prediction reintroduces the risk of inference privacy loss.

Homomorphic encryption (HE) [51] enforces privacy preservation by enabling computation directly on encrypted data. HE based protocols enable exact arithmetic operations over ciphertexts. Mathematical operations performed on ciphertexts produce encrypted results which, upon decryption, are equivalent to those obtained on plaintexts. This implies that the fusion center can process the data without accessing its raw form. While a subcategory of HE schemes theoretically support arbitrary computations, practical schemes are limited by ciphertext noise growth, requiring bootstrapping to refresh ciphertexts. Although HE ensures confidentiality, it does not inherently protect against inference attacks arising from ciphertext level patterns or correlations. Moreover, HE is computationally intensive, making it less suitable for resource constrained agents in MAS. Pulido et al. [60] outlined the challenges associated with HE based mechanisms. A primary issue is the dependency on a trusted third party authority (TPA) for key distribution. Such dependence on centralized trust is impractical in distributed MASs.

Efforts are being made to combine the strengths of DP and HE. For instance, Zhu et al. [84] used HE for secure model aggregation. However, the authors did not address the inference privacy attacks resulting from shared intermediate representations. Other works consider perturbations in quantized domains or rely on local differential privacy, but typically at the cost of slower convergence or biased updates.

From the perspective of coding theory and post quantum cryptography, lattice based constructions have gained prominence for their simplicity and security foundations [49, 58]. In distributed MASs, where agents share sensitive information, such constructions offer a promising foundation for privacy preserving communication and secure aggregation. Discrete Gaussian sampling over lattices underlies many cryptographic primitives and exhibits excellent concentration and closure properties that can be leveraged for privacy mechanisms. However, such schemes often incur significant computational and communication overhead, making them impractical for resource constrained agents. Furthermore, the large parameter sizes and implementation complexity associated with lattice based primitives may hinder scalability in distributed settings.

Hash functions, used predominantly for data integrity verification, can be used

to reduce the dimension of the data tuple, thereby attaining the inference privacy goal of MAS. However, the hash function chosen should be tunable with respect to the level of privacy desired by the agent. The design of cryptographic hash functions like SHA-256 [63], [61], Argon2 and Blake2 is such that a separate privacy parameter cannot be incorporated into it. Also, the complexity of algorithms like SHA-256, Argon2 and Blake2, render them unusable in the case of agents that have very limited computational power. Researchers [3] have explored the use of Bloom filters and choosing a hash function for the Bloom filter. Nevertheless, for MASs, we have to deploy a tunable hash function that would sanitize the data and would be incorporable in the agent. The principal characteristics and MAS-specific limitations of these methods are summarized in Table 2.2 for clarity and comparison.

Method	**Core Idea**	**Strength**	**Limitations in MAS**
Symmetric / Asymmetric Encryption [63]	Encrypt data before transmission using shared or public/private keys	Strong confidentiality during communication	Reveals full data upon decryption; does not preserve inference privacy
Paillier based Secure Aggregation [10]	Homomorphic addition enables secure aggregation without revealing individual inputs	Perfect confidentiality during aggregation	Expensive modular exponentiation; key management overhead; no inference privacy guarantee
Encrypted Neural Network Inference [78]	Integrates encryption and neural networks for secure computation	Enables inference over encrypted inputs	Decryption for prediction reintroduces inference privacy risk
Homomorphic Encryption [51]	Computation performed directly on ciphertexts	Allows processing without accessing raw data	High computational complexity; ciphertext noise growth; no protection from inference via patterns

Method	Core Idea	Strength	Limitations in MAS
Homomorphic Encryption Deployment Challenges [60]	Analysis of implementation challenges in distributed HE systems	Identifies key security and deployment considerations	Requires trusted third party for key distribution; unsuitable for fully decentralized MAS
Homomorphic Encryption-based Secure Model Aggregation [84]	Combines HE with distributed learning aggregation	Secure aggregation in collaborative learning	Does not address inference leakage from intermediate local agent representations that assist in training the global learning model
Ideal Lattice-Based Cryptography [49]	Uses lattice hardness assumptions for secure cryptographic construction	Post quantum security foundation	Large parameter sizes; significant computation and communication overhead
Lattice Cryptography Survey [58]	Framework for lattice-based encryption and secure computation	Strong theoretical security guarantees	Implementation complexity; scalability concerns in distributed MAS
Cryptographic Hash Functions (SHA-256) [63]	Framework for data integrity / data privacy	Strong collision resistance; widely standardized	No tunable privacy parameter; high computational cost for lightweight agents
SHA-256 Hash Function [61]	Cryptographic hashing for data integrity and fixed-length compression	Strong collision resistance; standardized design	No tunable privacy parameter; computationally heavy for lightweight agents

Method	Core Idea	Strength	Limitations in MAS
Bloom Filter based Mechanism [3]	Uses hash functions within probabilistic data structures for compact representation	Reduces data dimensionality; memory efficient	Privacy level not tunable; may leak information via collisions; limited inference protection

Table 2.2: Cryptographic and Coding-Theoretic privacy mechanisms and their limitations in MAS.

2.3 Reconstruction resistance via dimension reduction

This category of privacy preserving methods compresses the data tuple prior to communication. By compression, we imply a reduction in the dimension of the data tuple. This is done in a way that would retain the utility of the data as well as enforce privacy for the agent. Principal component analysis (PCA) [26, 48] is a classical example of such methods. In PCA, the directions of highest data variance are calculated, and the data components corresponding to those directions are retained. That is, the higher dimensional data is projected onto a lower dimensional subspace defined by the directions of greatest variance. This effectively reduces the number of features while preserving much of the original structure.

The other class of compression based methods, project the data tuples to a lower dimensional space by matrix multiplication. In such methods, a projection matrix is appropriately generated following certain conditions, and the raw data tuple is projected to the lower dimensional space using the generated matrix. The projection matrix plays a major role in the privacy-utility tradeoff making its choice crucial. The compressed representative is sent to the fusion center for aggregation and subsequent deduction of the system parameters.

Zhou et al. [82] proposed multiplying the entire data matrix with a matrix whose entries are drawn from a Gaussian distribution. In a follow up study [83], it was demonstrated that such a compressed representative retains enough information to deduce the system parameters with accuracy comparable to the original raw data.

A major drawback of the compression methods discussed, is the requirement

of a complete or a significant part of the dataset for its application. Agents in an MAS typically observe and transmit individual tuples in real time, without access to the full dataset, making these methods practically infeasible.

The requirement of a substantial part of the dataset is mitigated by basic random projection (BRP) [67]. BRP uses a fixed uniform random orthonormal matrix to project individual data tuples to a lower-dimensional subspace. This matrix serves as a basis for the lower-dimensional space. Although this method eliminates the need for the entire dataset, it introduces computational overhead due to the requirement of an orthonormal matrix, which is not computationally feasible for agents with limited processing capabilities. To consolidate the key insights from the above discussion, Table 2.3 provides a structured comparison of the existing methods and their respective tradeoffs.

Method	**Core Idea**	**Strength**	**Limitations in MAS**
Principal Component Analysis (PCA) [26]	Projects high-dimensional data onto directions of maximum variance	Preserves dominant structure; reduces dimensionality effectively	Requires access to full dataset; not suitable for real time tuple transmission
Improved / Modified Principal Component Analysis (PCA) [48]	Learns optimized principal subspace for dimensionality reduction	Improved utility preservation under compression	Dataset level computation required; not adaptive to streaming data
Gaussian Random Projection [82]	Multiplies data matrix with a Gaussian random matrix for dimensionality reduction	Theoretically preserves pairwise distances; simple linear transformation	Requires full data matrix; impractical for tuple wise MAS operation
Compressed System Identification [83]	Demonstrates that compressed Gaussian projections retain sufficient information for parameter estimation	Comparable estimation accuracy to original dataset	Still dependent on batch dataset; unsuitable for online MAS agents

Method	Core Idea	Strength	Limitations in MAS
Basic Random Projection (BRP) [67]	Projects individual tuples using a fixed orthonormal random matrix	Eliminates need for full dataset; supports tuple-wise projection	Requires orthonormal matrix generation; computational overhead for lightweight agents

Table 2.3: Compression Based Privacy Preserving Methods and their limitations in MAS.

2.4 Federated learning (FL)

Apart from the mechanisms discussed previously, another popular privacy mechanism in distributed inference is FL. In this approach, both the agents and the fusion center train a local and global model, respectively. Rather than transmitting the raw data after each observation, the agent transmits only the local model updates to the fusion center. However, naive FL is vulnerable to privacy leakage through gradients or model updates, which can be exploited to infer sensitive training data [53, 85].

Using FL, clients can collaboratively train a global model without sharing their raw data, thus maintaining an inherent level of privacy preservation [52]. Standard FL approaches such as FedAvg aggregate locally computed updates to train a global model in a communication efficient manner. Adversaries can exploit shared model updates to reconstruct private data using optimization-based gradient inversion attacks capable of high fidelity sample recovery [85, 33]. These findings underscore the necessity of robust and well designed privacy preservation mechanisms for practical FL deployments.

DP is known to limit information leakage during distributed training [1]. The differentially private stochastic gradient descent (DP-SGD)[39] algorithm enforces privacy by first clipping each per-example gradient to a fixed threshold. This process bounds the sensitivity of each client's contribution and provides a formal DP guarantee. However, selecting appropriate values for the clipping norm needs careful consideration, as static settings can lead to poor privacy-utility tradeoffs.

Existing studies have proposed adaptive approaches to mitigate this problem of privacy-utility tradeoff in DP-SGD. For instance, authors of [80, 32] adjust the clipping norm dynamically based on gradient statistics, while Galen et al. [6], scale the added noise according to the evolving privacy budget during training. Malekmohammadi et al. [50] further introduced a noise-aware aggregation frame-

work that explicitly accounts for client side heterogeneity in both data and noise levels. Their method estimates the effective noise variance of each client and re-weights model updates accordingly, achieving improved convergence and utility under heterogeneous privacy budgets. These adaptive mechanisms improve model utility relative to fixed DP configurations; however, they lack real time monitoring of privacy leakage and do not provide closed loop control based on empirical privacy risk.

Federated adaptive privacy preserving client level aggregation (FedAPCA) [72] proposes improvement over traditional DP, that addresses client heterogeneity by performing hierarchical aggregation with adaptive local DP across clients. However, it does not account for per timestep dynamics within individual agent signals. Other adaptive DP mechanisms in FL adjust privacy or noise allocation over time. Kiani et al. [46] propose time adaptive DP for learning rounds, while Hong et al. [41] and Zhang et al. [77] dynamically adjust noise variance or gradient clipping during optimization. These methods demonstrate the benefits of adaptive allocation, but focus on learning updates rather than raw streaming measurements. They also do not exploit the uncertainty of the signal per timestep. FL based methods are also computationally expensive, and need modifications for them to be applied to MAS settings.

Table 2.4 summarizes the discussed techniques and further illustrates the existing gaps that motivate the need for improved privacy mechanisms in MAS.

Method	**Core Idea**	**Strength**	**Limitations in MAS**
Federated Learning (FL) ([52])	Local model updates are shared instead of raw data	Reduces direct data exposure; inherent privacy preservation	Vulnerable to gradient inversion attacks; cannot prevent inference from shared updates
Gradient inversion attack ([53, 85, 33])	Adversaries reconstruct private data from shared gradients by optimizing gradient inversion for private data recovery	Highlights potential privacy risks in case of high-fidelity reconstruction attacks	Demonstrates susceptibility of naive FL to reconstruction attacks, thereby requiring additional privacy mechanism to counter inference breaches.

Method	Core Idea	Strength	Limitations in MAS
Differentially Private SGD (DP-SGD) ([1, 39])	Clips per-example gradients and adds noise to enforce DP	Enforces formal DP guarantees for distributed training	Fixed clipping norm may degrade utility; static DP does not handle streaming raw measurements
Adaptive gradient clipping ([80, 32])	Dynamically adjusts clipping norm using local gradient statistics	Improves convergence and privacy-utility tradeoff	No real-time monitoring of privacy leakage; limited feedback control
Dynamic noise scaling ([6])	Scales noise according to evolving privacy budget	Improves utility for heterogeneous clients	Focuses on noise scaling only; does not address per-timestep signal uncertainty
Noise-aware aggregation ([50])	Re-weights model updates based on effective noise variance per client	Handles heterogeneity; improves convergence and utility	Lacks real time privacy monitoring; computational overhead for MAS agents
Federated adaptive privacy preserving client level aggregation (FedAPCA) ([72])	Hierarchical aggregation with adaptive local DP per client	Accounts for client heterogeneity across agents	Does not handle per timestep dynamics; computationally expensive
Time adaptive DP ([46])	Adjusts privacy budget per learning round over time	Better tradeoff for sequential training	Focuses on training rounds, not the agent specific per timestep signals; does not exploit temporal uncertainty in raw data

Method	Core Idea	Strength	Limitations in MAS
Dynamic noise variance ([41])	Adjusts noise dynamically during optimization	Maintains DP and improves convergence	Only optimizes noise; raw streaming data signals not fully considered
Dynamic gradient clipping ([77])	Adjusts gradient clipping thresholds during optimization	Maintains formal DP; adapts to varying gradients	Computationally heavy; does not exploit per timestep signal uncertainty

Table 2.4: Federated Learning Based Privacy Mechanisms and their limitations in MAS.

2.5 Comparison of assumptions and privacy guarantees

The comparison in Table 2.5 highlights the differences amongst the surveyed paradigms with respect to their privacy guarantees but also in their practical assumptions and implementation requirements. Differential privacy provides a formal probabilistic privacy guarantee and is robust to arbitrary post-processing, whereas cryptographic methods primarily ensure confidentiality through computational hardness assumptions. Federated learning reduces raw data sharing but does not inherently provide privacy unless combined with mechanisms such as differential privacy or secure aggregation. In contrast, dimension reduction techniques generally achieve privacy empirically by reducing the recoverability of sensitive information while maintaining computational efficiency. Consequently, the choice of a privacy-preserving paradigm depends on the application requirements, adversary model, computational resources, and desired privacy guarantees.

Property	Differential Privacy	Cryptographic Methods	Federated Learning	Dimension Reduction
Trusted setup required	No	Often required	No	No
Presence of Randomness	Yes	Limited	Depends on computation of update	Depends on mechanism
Adversary	Inference adversary	Unauthorized adversary	Inference adversary	Inference adversary
Communication overhead	Low	High	High	Low
Per-step computation	Low	Moderate-High	Moderate-High	Low
Memory overhead	Low	Moderate-High	Moderate	Moderate
Privacy preserved under arbitrary post-processing	Yes	Yes	No	Depends on design
Robustness against auxiliary information	Formal guarantee	Depends on design	Depends on the incorporated privacy mechanism	Limited

Table 2.5: Comparison of privacy-preserving mechanism based on practical assumptions.

Alongside designing privacy mechanisms, an important problem in the MAS scenario is that of privacy-utility tradeoff. We now take a look at a set of optimization problems from the literature pertaining to the tradeoff problem.

2.6 Privacy-utility tradeoff problem

A classical approach to addressing this tradeoff problem is to formally define mathematical expressions for both utility and privacy, and pose the problem as one of constrained optimization [71]. Wang et al. [69] introduced a framework based on

the Cramér-Rao lower bound (CRLB) and the Fisher information matrix (FIM) to quantify utility and privacy. The worst case complexity of computing the CRLB is $\mathcal{O}(n^3)$ [40] which is computationally demanding for resource constrained agents.

The literature pertaining to this area of study can be further divided into subcategories according to the optimization objective. We either maximize utility subject to a fixed privacy loss, or maximize privacy subject to a required level of utility.

1. Maximizing utility: The optimization of utility can be achieved in two ways. The utility of the data exchanged can either be maximized or the utility loss minimized. In both cases, the privacy level required by the individual agents is fixed beforehand. There are two ways of viewing the problem.

   The authors in [69] pursue a sanitization function that would maximize the utility of the data. This implies that the optimum solution gives us a sanitization function that can be used to sanitize the agent data before being sent to the fusion center. Here the constraints for the optimization problem would be the individual privacy thresholds of the agents.

   The problem may also be reformulated in terms of utility loss. Wang et al. [68] consider the problem of minimizing utility loss in terms of the noise covariance matrix. Instead of solving for a sanitization function which would give the maximum utility, they try to find a noise covariance matrix which would result in the minimum loss of utility. The problem of finding the sanitization function which maximizes utility can be solved efficiently by considering the sequential method of solving optimization problems. However, solving the utility loss optimization problem is difficult because of the non linear relation between the utility function, the noise covariance matrix and the privacy thresholds.

2. Maximizing privacy: Contrary to the previous problem, here we assume that the privacy of the data is more important, hence a certain minimum utility level has to be met. Under this constraint, we maximize the privacy of each individual agent. The same problem can again be considered in two different ways.

   The maximization can be done over the set of all possible sanitization functions. This optimization problem with the constraint of perfect utility has been considered by Wang et al. [69]. The solution gives a sanitization function for which the privacy is maximized while ensuring perfect utility of the data being sent from agent to fusion center. The authors have further proposed a noise-addition based sanitization mechanism, named arbitrarily strong privacy with perfect utility sanitization algorithm (ASUP algorithm),

that has been shown to attain privacy under the constraint of preserving perfect utility. However, a significant limitation of their approach is its high computational cost.

The same optimization problem may also be considered with respect to the noise covariance matrix, that is, we try to find an optimum noise covariance matrix using which the desired goal will be achieved [71]. The privacy and utility functions, when defined using CRLB and FIM, are non-linear with respect to the noise covariance matrix. This implies that the problem is difficult to solve in this form. Reformulating the problem as a convex optimization one with linear matrix inequality constraints will allow us to solve it using semi-definite programming techniques.

3. Another class of the privacy-utility optimization problem tries to balance the tradeoff. Rather than maximizing one under preset values of the other, this set of problems tries to find an optimal sanitization function that would achieve a desirable utility and privacy tradeoff. The problem herein is to choose an appropriate privacy metric and then solving a non linear optimization problem. In [70], the authors use machine learning models to solve the problem. The choice of the privacy metric is crucial in such settings. The different data and inference privacy metrics are studied in [64, 65]. The authors define the various metrics as well as derive the interdependencies between the metrics.

4. The last category of optimization problems that we explore is that of minimizing the inference cost reduction of the adversary while ensuring that the utility loss of the data is below a certain threshold. A privacy mechanism is designed to keep the adversary from deducing either the raw data or the secondary data of an individual, entity or event. During privacy evaluation, designers try to breach the privacy to verify the resistance of the method against potential adversaries. When a part of the data is leaked to an adversary, the cost of inferring the rest of the data becomes relatively cheaper. This implies that there is a reduction in the cost of inference.

   The goal of this category of tradeoff problems is to minimize the reduction in inference cost. Duan et al. [20] consider this problem with the constraints of utility. That is, alongside the prevention of inference cost reduction, the utility of the data should not be lost.

All the optimization problems discussed thus far, use complex definitions of utility and privacy involving advanced statistical functions and non linear dependencies. This makes it hard to solve the problem computationally. As a consequence, these problems cannot be solved at the agent level, and would require

a third party dependency. Incorporating the solution in a privacy preserving algorithm increases the complexity of the algorithm. A summary of the tradeoff problems discussed thus far is given in Table 2.6.

Approach	**Objective**	**Strength**	**Limitations in MAS**
Utility maximization under fixed privacy ([69])	Find a sanitization function that maximizes data utility while satisfying predefined privacy thresholds	Provides optimal utility for given privacy; sequential optimization is feasible	Computationally intensive for resource constrained agents
Minimizing utility loss ([68])	Optimize noise covariance matrix to minimize loss of utility under fixed privacy constraints	Achieves minimal distortion of transmitted data	Non linear relation between utility, noise, and privacy thresholds makes optimization challenging
Privacy maximization under fixed utility ([69])	Optimize sanitization function to maximize privacy while preserving perfect utility	Ensures perfect utility while maximizing privacy; noise addition mechanism (ASUP algorithm) proposed	High computational cost; less scalable to large MAS
Privacy maximization via noise covariance ([71])	Optimize noise covariance matrix to achieve maximum privacy under utility constraints	Can be reformulated as convex optimization; solvable via semi-definite programming	Non linear CRLB/FIM functions make problem difficult; computational overhead remains high

Approach	Objective	Strength	Limitations in MAS
Optimal tradeoff between utility and privacy ([70])	Solve non-linear optimization to achieve a balanced privacy-utility tradeoff using appropriate privacy metric	Allows flexible balance; can incorporate different privacy metrics	Choice of privacy metric critical; solving non linear optimization may be computationally expensive
Privacy-utility analysis via interdependent metrics ([64, 65])	Study relationships between data privacy and inference privacy metrics to guide tradeoff optimization	Provides deeper understanding of privacy interdependencies; useful for designing metrics-aware mechanisms	Metrics computation and dependency analysis may be complex for MAS
Minimizing adversary inference cost reduction ([20])	Design mechanism to prevent adversaries from reducing inference cost while maintaining data utility	Ensures strong inference privacy protection alongside utility preservation	Requires adversary modeling; optimization may be computationally heavy

Table 2.6: Brief summary of the privacy-utility tradeoff problems.

2.7 Research gaps and motivating questions

The literature review highlights several challenges in privacy-preserving MASs. These challenges motivate the research presented in this thesis and are summarized as the following research questions.

- **RQ1:** How can inference privacy be preserved in decentralized multi-agent systems without relying on trusted centralized entities?
- **RQ2:** In what way can privacy mechanisms take into account the spatio-temporal correlations present in streaming observations?

- **RQ3:** How can computationally lightweight privacy mechanisms be designed to uphold privacy leakage bounds while enhancing utility?
- **RQ4:** Are privacy mechanisms capable to adapt their protection levels according to varying data sensitivity and application requirements?
- **RQ5:** What reconstruction-resistant data representations can be formed that would preserve performance while minimizing sensitive information leakage?

The algorithms proposed in the subsequent chapters are developed to address these research questions through privacy-preserving mechanisms.

Part I

Perturbation Based Methods

Chapter 3

Differential privacy for mobile agents with continuous queries

This chapter establishes the short-comings of uniform budget allocation based differential privacy mechanisms in case of mobile agents with continuous queries. To mitigate the limitation, an efficient perturbation generation algorithm is introduced using the mobility and spatio-temporal profile of the agents [1].

[1]The contents of this chapter appear in the paper "On the effectiveness of differential privacy to continuous queries". **Puspanjali Ghoshal, Mohit Dhaka, Ashok Singh Sairam**. Service Oriented Computing and Applications (SOCA) 18, 381–395, Springer (2024).

In most MASs, agents repeatedly communicate with the fusion center. In particular, MAS models providing location based services (LBSs) involve individual agents sending their location data continuously over a period of time to get the necessary service. The information sent by an agent from adjacent locations over a span of time is referred to as continuous queries. For example, a man moving on the road searches for nearby points of interest (PoIs), say restaurants. The setup is shown in Figure 3.1. The user fires the query from locations (x_1, y_1), (x_2, y_2) and (x_3, y_3) at time instants t_1, t_2 and t_3, respectively. Suppose the user uses differential privacy and the corresponding perturbed locations advertised by the user are (x_{p1}, y_{p1}), (x_{p2}, y_{p2}) and (x_{p3}, y_{p3}). The individual perturbations appear to be seemingly protected. However, if we consider the user's mobility pattern, the road constraints, and the query content, the privacy level of the user will be reduced. Thus, a predominant constraint in continuously evolving location data is the spatio-temporal correlation. The mobility profile and movement history accompanied by the road layout can be exploited to carry out inference privacy breaches. To avoid such attacks, an agent has to use a decentralized privacy-preserving mechanism.

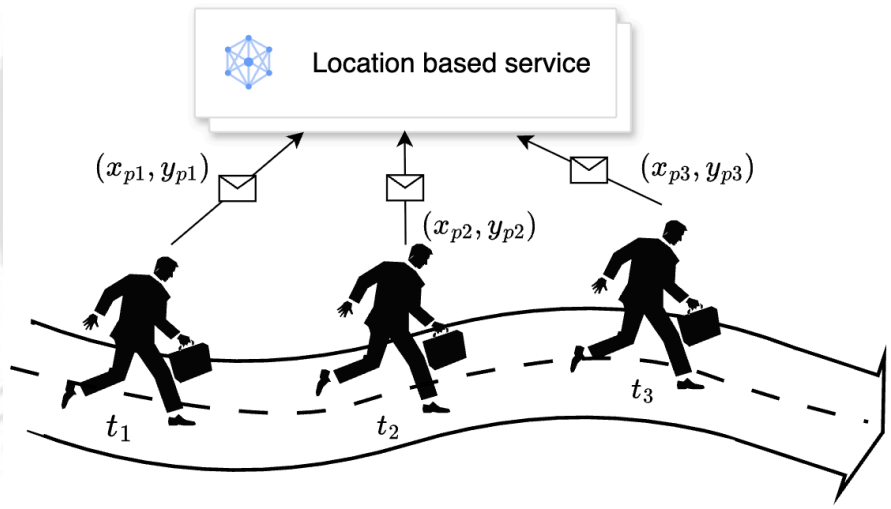


Figure 3.1: Illustration of continuous queries.

Differential privacy (DP) based methods [24], which were initially introduced for statistical data, have found wide applicability in LBSs too. However, the standard DP methods, with uniform per timestep budget allocation, lead to a reduced privacy level over time due to sequential composition of the budgets. Assuming ϵ to be the privacy level for applying noise independently to a location, in this chapter, we prove that the privacy level is reduced to $n'\epsilon$, when applied to n' consecutive locations. Further, we propose a perturbation-based privacy framework for handling continuous queries. The framework incorporates a DP perturbation mechanism as a fallback and empirically performs better than standard DP mech-

anisms. The results are validated empirically using a real dataset.

The chapter is organized as follows. We describe the system model for continuous queries in Section 3.1. In Section 3.2, we introduce the definition of DP and how the classical definitions are correlated to our problem domain. Next, we formally define the problem statement from the attacker's point of view in Section 3.3 followed by the proposed perturbation framework in Section 3.4. The experimental evaluation is detailed in Section 3.5.

3.1 System model for continuous queries

In this chapter, for the application of DP, we assume that the area of interest is divided into uniform rectangular grids [75]. This discretization is introduced to obtain a finite, well-structured domain that simplifies adjacency definitions, probability computations, and practical implementation. Each location (grid cell) is uniquely identified by a pair of longitude and latitude, as illustrated in Figure 3.2. If the latitude and longitude are x and y respectively; then the corresponding cell number is

$$x + (y - 1) * N_x. \tag{3.1}$$

Here N_x is the number of partitions in latitude (which in the case of Figure 3.2 is 5). The privacy-preserving mechanism $\mathcal{M}$ deployed by the user, perturbs each location independently. Let $\mathcal{S}$ be the set of all possible locations for the user. More precisely $\mathcal{S}$ is the set of all cells in the space grid.

Only the user knows the actual cell location and the perturbed locations are reported to the service provider. However, background knowledge can be exploited by an adversary to postulate the following probabilities.

Definition 3.1 *(**Residing probability**) The residing probability $P_t(i)$ is the probability of a user being in cell s_i at a given time instant t.*

That is, if we consider a user u, the probability that it will reside in cell s_i at time t is $P_t(i) = Pr(u_t = s_i)$. For example, $P_2(7)$ is the probability that the user is in cell s_7 at time $t = 2$. We can denote the user's probability distribution in space at any time t by $P_t = P_t(1)$, $P_t(2)$, ..., $P_t(N)$, where $\sum_{i=1}^{N} P_t(i) = 1$.

Definition 3.2 *(**Perturbation probability**) Perturbation probability (G) is the probability of a cell s_j being generated as a perturbed location with respect to the actual cell location s_i. We use a matrix G[i][j] to denote the probability.*

We can define perturbation probability as G[i][j] = $Pr(\mathcal{M}(s_i) = s_j)$, where $\mathcal{M}$ is the privacy-preserving mechanism. As this probability is derived from the privacy-preserving mechanism, it is considered to be known to the adversary. For example,

5	$s_{21}=21$	$s_{22}=22$	$s_{23}=23$	$s_{24}=24$	$s_{25}=25$	
4	$s_{16}=16$	$s_{17}=17$	$s_{18}=18$	$s_{19}=4+(4-1)*5=19$	$s_{20}=20$	
3	$s_{11}=11$	$s_{12}=12$	$s_{13}=13$	$s_{14}=14$	$s_{15}=15$	
2	$s_6=6$	$s_7=7$	$s_8=3+(2-1)*5=8$	$s_9=9$	$s_{10}=10$	
1	$s_1=1+(1-1)*5=1$	$s_2=2$	$s_3=3$	$s_4=4$	$s_5=5$	
	1	2	3	4	5	

Figure 3.2: Space division with cell numbers.

in Figure 3.2, G[1][2] is the probability that the user is in cell 1 and the perturbed location is cell 2.

Definition 3.3 *(**Transition probability**) Transition probability (M_p) is the probability that a user in cell s_i at time instant t will move to cell s_j in the next time instant. We use a matrix $M_p[i][j]$ to denote the probability.*

Consider P_t and P_{t+1} to be the residing probabilities of the user at time t and $t+1$, respectively. The relation between residing and transition probability is given as $P_{t+1} = P_t \mathrm{M}_p$. For example, in Figure 3.2, $\mathrm{M}_p[1][2]$ denotes the probability that the user moves from cell 1 to cell 2 in one unit of time.

Definition 3.4 *(**Trace**) The trace of a user is the set of consecutive locations that a user visits.*

Trace of a user is expressed as an n'-tuple if the user visits n' locations consecutively. For example, if the consecutive locations of a user at time instants t_1,t_2, ...,$t_{n'}$ are a_1,a_2, ...,$a_{n'}$; then the trace of the user is represented as $\mathfrak{T}=(a_1, a_2, ..., a_{n'})$.

3.1.1 Evolution of residing probability

Let pr_t^- be the residing probability of the user at time t before observing the perturbed location (prior probability). Analogously, let pr_t^+ denote the residing probability after observing the perturbed location (posterior probability). From the definition, we can say $pr_{t+1}^- = pr_t^+ \mathrm{M}_p$, that is, we can calculate the prior probability at time $t+1$ using the transition matrix and posterior probability at time t. In general, we can compute the posterior probability using the prior probability and the Bayes' theorem[75].

$$pr_t^+[j] = Pr(u_t = s_j | o_t) = \frac{Pr(o_t | u_t = s_j) * pr_t^-[j]}{\sum_i Pr(o_t | u_t = s_i) * pr_t^-[i]}. \tag{3.2}$$

The prior probability denotes one of the possible previous locations of the user. From the adversary's viewpoint, the perturbed location (posterior probability) and the mobility profile are known. Thus, it can use the hidden Markov model to compute the prior probabilities.

3.1.2 Problem statement

In this chapter, we consider that the user is mobile, so it reports the perturbed position of n' locations to the service provider instead of a single location. Let $\mathfrak{T} = (a_1, a_2, \ldots, a_{n'})$ be the actual locations of the user, that is, the actual trace of the user. Let $O = (o_1, o_2, \ldots o_{n'})$ be the corresponding perturbed locations reported by the user. We call this the perturbed trace. The adversary can perform a back calculation and find the set of all possible traces $\mathcal{L}$ from the observed n' perturbed locations; that is $\mathfrak{T} \subseteq \mathcal{L}$. The task of the adversary will be to scan through $\mathcal{L}$ and map the perturbed trace to the actual trace.

We define the problem statement from the adversary's viewpoint. The task of the adversary is to maximize the privacy breach. Let n' be the trace size and $\mathfrak{T}$ be the set comprising the actual n' locations of the user. Let O be the observed perturbed trace. With this information, the adversary will estimate the actual user locations. We denote the adversary's conjecture as $\mathfrak{T}' = (a'_1, a'_2, \ldots, a'_{n'})$. The problem is to minimize the distance between $\mathfrak{T}$ and $\mathfrak{T}'$.

$$\min \left(\sum_{i=1}^{n'} d_x(a_i, a'_i) \right), \tag{3.3}$$

where d_x is a distance function.

3.2 Preliminaries

The privacy definitions used in this thesis are based on the classical definition of DP applied to an arbitrary set of secrets. In the case of LBS, the user's location is considered a secret. This section discusses DP by introducing the basic definitions and its enforcement in LBS.

3.2.1 Standard DP mechanism

The concept behind DP is to perturb aggregate functions over a database so that adding or removing one record to the database should not reveal any information about that entry. At the same time, the perturbation should not decrease the accuracy of data substantially. For example, if a survey tells us about the number of people having cancer in a city then DP ensures that multiple organizations can release statistical aggregate data without revealing sensitive information about the participants. Due to the ability of DP to provide a guarantee of privacy protection, it is increasingly used to protect sensitive information of an individual, such as user's events and real time location. The application of DP requires identifying neighbouring datasets. In the context of location privacy, the data will be a set of locations. We now formally define neighbours and neighbourhood in the context of LBS.

Definition 3.5 *(**Neighbourhood**) Two locations pertaining to a user are said to be neighbours if they lie in adjacent cells.*

In the context of an LBS, a dataset is a collection of locations which we call as trace. In this chapter, we use the term dataset and trace interchangeably. Our description of neighbourhood is different from the classical definition, where two datasets are said to be neighbours if one dataset is a subset of another and the larger dataset contains exactly one additional entry. The formal definition of neighbouring traces is presented in Definition 3.8.

Let $\mathcal{M}$ be a privacy mechanism that satisfies ϵ-DP. It means that $\mathcal{M}$ is a randomization function such that the addition or deletion of a single input element does not change the probability of the output by more than e^{ϵ}. An important property of DP is that an adversary cannot gain additional information about the secret from the perturbed data. However, this may not be true in the case of LBS, since the adversary can use background knowledge of the user such as his mobility history to gain more insight of the private data.

Definition 3.6 *(**Differential privacy**) Any privacy mechanism $\mathcal{M}$ provides ϵ-DP if, for any pair of neighbouring datasets D_1 and D_2 and for any $O \in Range(\mathcal{M})$,*

$\mathcal{M}$ *satisfies*

$$Pr[\mathcal{M}(D_1) = O] \leq e^{\epsilon} Pr[\mathcal{M}(D_2) = O].$$

The parameter ϵ is assumed to be a preset characteristic, which is a measure of the privacy level attained by the mechanism. By appropriately tuning this parameter, one can control the privacy level and consequently the noisiness of the perturbed data. Smaller ϵ value guarantees better privacy with noisier results. Another parameter used to control the noise level of the results is sensitivity. It measures the largest possible difference of the perturbed data tuples. Larger sensitivity yields noisier results.

When we have a sequence of mechanisms which provide DP independently, the final privacy is the sum of the individual privacy. For example, consider that mechanism $\mathcal{M}_i$ provides ϵ_i DP. In case we apply a sequence of these mechanisms on a dataset D, the final privacy is $\sum \epsilon_i$ DP.

Definition 3.7 ***(Global sensitivity)*** *Global sensitivity for a mechanism* $\mathcal{M}$ *is defined as the norm of maximum difference between output of two neighbouring datasets i.e.*

$$\Delta q = \max_{D_1, D_2} ||\mathcal{M}(D_1) - \mathcal{M}(D_2)||.$$

Here, Δq characterizes the scale of the influence of one data point, and hence how much noise needs to be added. To achieve ϵ-DP, the original location is perturbed using a random noise that is generated from a distribution calibrated with both privacy budget ϵ and Δq.

3.2.2 Laplace Distribution

DP can be achieved by adding a symmetric geometric noise to the true location. The Laplace mechanism naturally generalizes such a geometric mechanism. DP acquires its strength from the fact that it draws noise from the Laplacian distribution, a symmetric exponential distribution. The Laplace function to achieve ϵ-DP is:

$$f(x) = \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}, \tag{3.4}$$

where $\lambda = \frac{\triangle q}{\epsilon}$.

3.2.3 Laplace mechanism for DP

Suppose we have a query function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^d$, which maps the database to d real numbers and the desired privacy threshold is ϵ. The noise is drawn from

the Laplacian distribution and is added coordinate-wise to the query result. The Laplace mechanism can be written as follows:

$$\mathcal{M}(f, X, \epsilon) = f(X) + (Y_1, \ldots, Y_d); \tag{3.5}$$

where $(Y_1, ..., Y_d)$ are independent and identically distributed random variables drawn from $Lap(\frac{\Delta f}{\epsilon})$,

$$X \in \mathcal{X};$$

$$\Delta f = \max_{\substack{x,y \in \mathbb{N}^{|\mathcal{X}|} \\ ||x-y||_1 = 1}} ||f(x) - f(y)||[24].$$

3.2.4 DP for point queries

Point queries are assumed to be independent and the exact user coordinates are communicated along with the query. To use DP for handling point queries involving location data, noise is added to the queries in such a way that the perturbed location cannot be related to any particular user from a set of users [15, 44]. We consider $\mathcal{M}$ as the mechanism which perturbs the coordinates of the user. The x and y coordinates are perturbed independently.

Let us assume a set of k users whose coordinates are $[(x_1,y_1), (x_2,y_2), ..., (x_k,y_k)]$. Let us define (x_p,y_p) as the perturbed location corresponding to true location, (x_t,y_t). The perturbed location (x_p, y_p) is generated such that for any two locations (x_i, y_i) and (x_j, y_j) from the set,

$$\begin{aligned} Pr[x_i \to x_p] &\leq e^{\epsilon} Pr[x_j \to x_p] \\ Pr[y_i \to y_p] &\leq e^{\epsilon} Pr[y_j \to y_p], \end{aligned} \tag{3.6}$$

where $\epsilon > 0$ and $i, j \in \{1, \ldots, k\}$.

From Section 3.2.3, we know that the Laplace distribution can be used to achieve this property with the scale parameter $\lambda > 0$. The perturbations are generated such that:

$$\begin{aligned} Pr[x_i \to x_p] &= \frac{1}{2\lambda} e^{\frac{|x_i - x_p|}{\lambda}} \\ Pr[y_i \to y_p] &= \frac{1}{2\lambda} e^{\frac{|y_i - y_p|}{\lambda}}, \end{aligned} \tag{3.7}$$

where $\lambda = \frac{\Delta q}{\epsilon}$, and

$$\Delta q = \begin{cases} \max x_i - \min x_i, & \text{for x coordinate} \\ \max y_i - \min y_i, & \text{for y coordinate.} \end{cases}$$

In case of LBS, an adversary can use the mobility profile of the user to get

secondary information about the private data. Another drawback of noise based DP mechanisms is the fixed noise levels and requirement of higher noise for higher privacy values. The fixed noise levels do not take into consideration the informativeness of the data at the instant and a higher noise level at each stage would result in degraded data utility.

3.2.5 DP for continuous queries

Having established the standard differential privacy framework for point queries, we now examine its application to continuous queries. While independently perturbing each location using the Laplace mechanism provides differential privacy at every time instant, the accumulated disclosures across time remain vulnerable to inference attacks due to spatio-temporal correlations. This limitation motivates the perturbation framework proposed later in this chapter, which is designed to improve inference privacy and uses a DP based method only as a fallback.

At this point, we assume that a user instead of sending a single location to the service provider, communicates a set of n' locations. Let $x = \{x_1, \ldots, x_{n'}\}$ denote the x-coordinates of the trace. Similarly, let $y = \{y_1, \ldots, y_{n'}\}$ denote the y-coordinates. For the sake of generalization, we assume the locations are adjacent (neighbouring data points). The user uses Equation 3.7 to perturb the location at each time instant. The corresponding perturbed locations shared by the user will appear as $(x_{p1}, y_{p1}), \ldots, (x_{pn'}, y_{pn'})$. Although the individual perturbations considered independently will have ϵ privacy, there is a strong spatio-temporal correlation between the data points.

In order to evaluate the efficacy of a privacy preserving mechanism, we first need to comprehend the adversary's knowledge. In this era of data explosion, an attacker can gather background knowledge through various means: publicly available data aggregation, observing prior events, historical data and so on. We formulate the problem from an attacker's viewpoint.

3.3 Privacy evaluation

Given the perturbed locations of trace size n', the aim of the adversary is to estimate the actual location for each of the observed traces. We can write this mathematically as

$$\prod_{i=1}^{n'} Pr(a_i = a'_i | \mathcal{M}(a_i) = o_i). \tag{3.8}$$

Using Bayes' theorem, we can rewrite the above equation as

$$\prod_{i=1}^{n'} \frac{Pr(\mathcal{M}(a_i) = o_i | a_i = a'_i) * Pr(a_i = a'_i)}{\sum_x Pr(\mathcal{M}(a_i) = o_i | a_i = x) * Pr(a_i = x)}. \tag{3.9}$$

To make our computations simpler, in Equation 3.9, we consider all the candidate locations (x) to be equally probable to be the user's actual location. That is, $Pr(a_i = x) = Pr(a_i = a'_i)$. Substituting this equality, we can rewrite the equation as follows

$$\prod_{i=1}^{n'} \frac{Pr(\mathcal{M}(a_i) = o_i | a_i = a'_i)}{\sum_x Pr(\mathcal{M}(a_i) = o_i | a_i = x)}. \tag{3.10}$$

The denominator of Equation 3.10 is the sum of all the probable locations. Thus it will equate to 1 and the equation will be reduced to

$$\prod_{i=1}^{n'} Pr(\mathcal{M}(a_i) = o_i | a_i = a'_i). \tag{3.11}$$

The above equation is the probability of generating o_i as the perturbed location, given a_i is the actual location. Further, we assume that the estimated location is the same as the actual location. Thus using Definition 3.2, we can rewrite Equation 3.11 as

$$\prod_{i=1}^{n'} G[a'_i][o_i]. \tag{3.12}$$

In the case of continuous queries, the adversary needs to estimate the actual location of not only the current perturbed location but also for all the consecutive locations traversed by the user. The attacker can use the transition probability defined in Definition 3.3 to capture the user movement. Mathematically, we can express it as

$$\prod_{i=1}^{n'-1} M_p[a'_i][a'_{i+1}]. \tag{3.13}$$

The expression in Equation 3.12 denotes the probability of generating the perturbed location from the estimated actual location. The expression in Equation 3.13 denotes the probability of generating the actual locations of the trace. Thus using both these equations, we can express the objective of the adversary as an optimization problem in terms of the given probability matrix.

$$\max((\prod_{i=1}^{n'-1} M_p[a'_i][a'_{i+1}]) * (\prod_{i=1}^{n'} G[a'_i][o_i])). \tag{3.14}$$

The attacker will use the above equation to find the best possible trace. The steps are outlined in Algorithm 1. The first step is to find $\mathcal{L}$, the set of all the possible traces of size n'. To compute $\mathcal{L}$, we do not consider all the traces, but an area within which the actual location of the user will be contained. The area can be computed by taking into consideration the mobility profile and the perturbed trace of the user. From among these candidate traces we choose the best one by solving the maximization problem given in Equation 3.14.

Algorithm 1 Algorithm to compute actual trace from perturbed trace

Input: Perturbed trace O
Output: Possible Actual trace ($\mathfrak{T}'$)

$\mathfrak{T}' \leftarrow [\,]$
max_prob $\leftarrow 0$
$\mathcal{L} \leftarrow$ Compute all possible traces of size n'
for all $X \in \mathcal{L}$ **do**
    curr_prob $\leftarrow$ [Compute probability using Equation 3.14]
    if (curr_prob > max_prob) **then**
        $\mathfrak{T}' \leftarrow X$
        max_prob $\leftarrow$ curr_prob
    end if
end for

3.3.1 Evaluation of continuous queries

Let $\mathcal{M}$ be a DP mechanism that guarantees ϵ privacy for point queries. As stated in Section 3.2, the parameter ϵ is a measure of the loss of privacy due to $\mathcal{M}$. We want to perform a worst case analysis of the privacy mechanism for continuous queries of trace size n'. That is, we want to quantify the level of privacy that is compromised. We make the assumption, that $\mathcal{M}$ is applied independently to each of the locations. We show that the privacy level is reduced to $n'\epsilon$. That is, the privacy level of $\mathcal{M}$ decreases linearly with respect to the trace size. To prove our hypothesis, first we define neighbourhood traces.

Definition 3.8 ***(Neighbouring trace)*** *Two traces $\mathfrak{T} = \{a_1, \ldots, a_{n'}\}$ and $\mathfrak{T}' = \{a'_1, \ldots, a'_{n'}\}$ are called neighbours if locations on the same index are neighbours i.e. a_i and a'_i are neighbours, for all $i \in$ [1,n'] (according to Definition 3.5).*

Thus, the adjacency relation adopted in this chapter is, in turn, based on bounded geographic distance between corresponding locations in two traces.

Theorem 3.1 *An ϵ-DP mechanism when applied to continuous queries of trace size n', the cumulative privacy loss increases to $n'\epsilon$.*

Proof. Let $\mathcal{M}$ be the location privacy preserving mechanism and $O = \{o_1, \ldots, o_{n'}\}$ be the observed trace. Further, let $\mathfrak{T} = \{a_1, \ldots, a_{n'}\}$ and $\mathfrak{T}' = \{a'_1, \ldots, a'_{n'}\}$ be any two neighbouring traces. Using the definition of DP, we can write the relation between any two points a_i and a'_i as

$$Pr[\mathcal{M}(a_i) = o_i] \leq e^{\epsilon} * Pr[\mathcal{M}(a'_i) = o_i].$$

We extend this definition for the entire trace A

$$\begin{aligned} Pr[\mathcal{M}(\mathfrak{T}) = O] &= \prod_{i=1}^{n'} Pr[\mathcal{M}(a_i) = o_i] \\ &\leq \prod_{i=1}^{n'} e^{\epsilon} Pr[\mathcal{M}(a'_i) = o_i] \\ &\leq e^{n'\epsilon} \prod_{i=1}^{n'} Pr[\mathcal{M}(a'_i) = o_i] \\ &\leq e^{n'\epsilon} Pr[\mathcal{M}(\mathfrak{T}') = O]. \end{aligned}$$

Using the same argument, we can prove that

$$Pr[\mathcal{M}(\mathfrak{T}') = O] \leq e^{n'\epsilon} Pr[\mathcal{M}(\mathfrak{T}) = O].$$

□

The above result shows that repeated independent application of an ϵ-DP mechanism results in a cumulative privacy-loss bound of $n'\epsilon$ under sequential composition. That is, under sequential composition, the composed mechanism satisfies $n'\epsilon$-differential privacy. Motivated by this increasing cumulative privacy loss for long traces, we now introduce the proposed perturbation mechanism.

3.4 Perturbation based privacy framework for continuous queries

There are two ways in which we can build a privacy framework to handle continuous queries: mobility profile-based approach [75] and prediction-based approach [13]. In the mobility profile-based approach, we use the previous true locations of the user to generate perturbed locations instead of generating them independently. On the other hand, the prediction-based approach uses the previously reported locations to generate a new perturbed location. In this chapter, we propose a privacy mechanism that is a fusion of the two approaches. The proposed mechanism exploits the neighbouring locations to generate the perturbed location.

At the same time, it allows the user to specify the level of utility desired. We resort to the standard noise addition mechanism only if we fail to find a feasible location. The algorithm for the proposed privacy mechanism is given in Algorithm 2.

Algorithm 2 Algorithm for generation of perturbed location

Input: Actual location $\mathfrak{T}$
Output: Predicted perturbed location O
$X \leftarrow \langle$ Mobility profile of user$\rangle$
$pr_t^-(i) \leftarrow$ prior_probability(X, pr_{t-1}^+) ▷ Compute prior prob.
ΔX $\leftarrow$ δ-set(X,$pr_t^-(i)$) ▷ Compute delta set using Eq. 3.15
Threshold $\leftarrow \langle$ User customized distance value $\rangle$
for all $y \in \Delta X$ **do**
    if $d(y, \mathfrak{T}) <$ Threshold **then**
        $\Omega[] \leftarrow y$
    end if
end for
if $\Omega \neq \emptyset$ **then**
    pred_loc $\leftarrow y = g(\Omega)$ ▷ Select a location from Ω randomly using g
else
    $O \leftarrow \mathfrak{T} + Noise$
end if
Calculate revised posterior probabilities using Mobility profile
return O

The proposed perturbation based privacy framework can be broadly divided into three stages. In the first step, we find the neighbouring locations of a user. The second step examines the predicted locations and returns those positions that can be used as a perturbed position. One of the feasible locations is reported as a perturbed location. If the test fails, we employ a noise mechanism to perturb the actual location. A brief outline of the steps involved is given in the following subsections.

3.4.1 Generating neighbouring locations

The basis of DP is to hide the true location in neighbouring locations. In statistical databases, neighbouring databases are obtained by adding or deleting a record, but this approach is not possible for spatio-temporal databases. We use the notion of δ-location set [75] to find the neighbour locations. At a given time instant t, it is a set containing the minimum number of locations such that the sum of their prior probability is at least equal to $1 - \delta$. The parameter δ indicates the possible locations to be included. A value of δ equals 0 means including all possible locations. The advantage of using this set is that the locations a user frequents are protected; the number of locations is also low, thereby avoiding significantly high perturbation. If X is the mobility profile of the user, the formula to compute

the δ-location set is reproduced in Equation 3.15.

$$\delta\text{-}set(X) \leftarrow \min\{a_i \mid \sum_{a_i} p_t^-(i) > 1 - \delta\}. \tag{3.15}$$

3.4.2 Examine the feasibility of the neighbouring locations

Having generated the neighbouring locations, we need to check their feasibility to be usable as a perturbed location. We employ a testing mechanism to check the quality of service achieved for each position in the set. We find the distance $d(y, \mathfrak{T})$ between the locations y in the δ-location set and the actual user location $\mathfrak{T}$. All the locations satisfying a threshold criterion are collected in a set Ω. The threshold is set considering the desired level of utility.

The choice of distance metric will depend on the characteristic of the data points. The Euclidean distance measures the shortest distance and works well for data points that are dense and continuous. On the other hand, if the data points are locations on a road map, the Manhattan distance will be more appropriate. It measures the *taxi* distance, that is the distance between two points in a city while moving on a uniform grid, like city blocks. In this work, we have used the Euclidean distance.

After testing, if there are multiple locations in the set, we choose one of them randomly as the predicted perturbed location. On the other hand, if none of the neighbouring points satisfies the threshold criterion, we add noise to the actual user location to generate a new perturbed location.

3.4.3 Noise addition mechanism

The algorithm resorts to noise addition if the set of feasible neighbouring locations is a null set. In such a scenario, the coordinates of the location are perturbed by noise as in the case of standard DP. The method of generation of perturbed data using noise addition can be achieved by using methods like the Laplacian mechanism and Gaussian mechanism [24]. The Laplacian method has been introduced in Section 3.2.3. Alternatively, Gaussian mechanism can be used which draws the noise from a Gaussian distribution. Coordinate-wise addition of noise has proven to give the DP guarantee.

The objective of the proposed framework is to improve inference privacy preservation for continuous location queries by exploiting mobility information and neighbouring feasible locations. The proposed framework is not intended to satisfy DP as a standalone mechanism. DP is used only as a fallback when no feasible neighbouring location is available.

3.5 Experiment and results

In this section, we present the experimental results to corroborate our theoretical analysis. The experiments are performed using the privacy evaluation framework described in Section 3.3. Specifically, the adversary estimates the user's actual trace by solving the optimization problem in Equation 3.14, which combines the perturbation probability matrix and the mobility profile transition matrix. As we use probability distributions, there is marginal variation in the results of an experiment when repeated. We, therefore, run each experiment 100 times, and the average value of the outputs is considered the final result.

3.5.1 Datasets

We experimented with the following real-world dataset:

- Geolife data [81]: The dataset is from Microsoft Geolife GPS trajectory research data. The dataset contains the trace of about 180 users collected over a period of three years. We used 12000 locations to carry out the experiment. The mobility profile M_p of the user is calculated using the user trajectory database and is used to estimate the transition probabilities between successive locations during the inference process.

3.5.2 Experimental setup

The geographical area is discretized into a grid, where each grid cell represents a possible user location. The adversary is assumed to know the released perturbed locations and the user's mobility profile. Using the mobility profile, the adversary estimates the user's current location from the observed trajectory. The prior probabilities are computed from the fitted Beta distribution together with the posterior probabilities from the previous time instant, thereby capturing the sequential nature of user mobility.

3.5.3 Performance metrics

A user who reports his exact location gets a hundred percent service but no privacy. On the other hand, if a user reports a location very far from his actual location, he gets full privacy but no service. Thus, all privacy mechanisms can be viewed as a trade-off between privacy and quality of service. In our experiment, we use two metrics each to measure privacy and the quality of service. To compute the level of location privacy provided, we introduce a metric breach count. It gives a measure of the number of times an adversary can correctly guess the user's actual

location. The second metric is drift ratio, a measure of the number of times the true location falls outside a predefined location set. To measure the quality of service (QoS), we use two other metrics, displacement and resemblance. Both these metrics have been adapted from the work by Dewri et al. [15]. The metrics are dependent on the capability of the adopted attacker to estimate the user's true location and is interpreted as an empirical attack-specific measure.

3.5.3.1 Breach count

Breach count is the average number of locations guessed correctly by the adversary. We say that an attacker can locate the user if he is able to compute the cell of the user correctly. Let I be an indicator variable that is set to 1 if the attacker correctly conjectures the user's exact location. Otherwise, the variable is set to 0. For a given estimated trace $\mathfrak{T}'$ and an actual trace $\mathfrak{T}$ of size n', the breach count is defined as :

$$\text{breach count} = \frac{\sum_{i=1}^{n'} I_{(a_i = a_i')}}{n'}.$$

3.5.3.2 Drift ratio

We compute the δ-location set using the definition given in Section 3.4.1 for each timestamp. Enumerating the set requires computing the prior probabilities. We realize that our data follows a Beta distribution. Using this distribution, for each timestamp, we find their prior probabilities using the posterior probability of the previous timestamp and the Mobility profile. The value of δ is set to 0.1, and accordingly, the delta location set is computed.

By definition, the δ-location set does not guarantee to include the actual location. The ratio of times the actual location falls outside the δ-location set (drift) to the total number of time stamps is defined as drift ratio.

3.5.3.3 Displacement

Displacement [15] is defined as the average distance between the actual and perturbed locations in the trace. For a given estimated trace $\mathfrak{T}'$ and an actual trace $\mathfrak{T}$, the displacement is defined as

$$\text{displacement} = \frac{\sum_{i=1}^{n'} \rho(a_i, a_i')}{n'}.$$

The separation between two points is measured in terms of the Euclidean distance (i.e. the l^2 norm $||.||_2$). Let (x_i, y_i) and $(x_i{}', y_i{}')$ be the coordinates of a_i and a_i', respectively. The distance between these two points is given as $\rho(a_i, a_i') = ||a_i - a_i'||_2$.

3.5.3.4 Resemblance

Resemblance [15] gives a measure of the PoIs that are common in the query using actual and perturbed location. More formally, let $N_R = \{s_1, \ldots, s_k\}$ be the set of k nearest neighbours corresponding to the actual location. Similarly, let $N'_R = \{s'_1, \ldots, s'_k\}$ be the PoIs corresponding to the perturbed location. Then resemblance is defined as

$$\frac{|N_R \cap N'_R|}{|N_R|}.$$

The metrics presented in this section are empirical evaluation measures that quantify the practical reconstruction risk and utility of the released locations under a specific attack model. Consequently, these metrics act as complementary indicators for evaluating the effectiveness of the proposed mechanism against the chosen adversary.

3.5.4 Empirical results

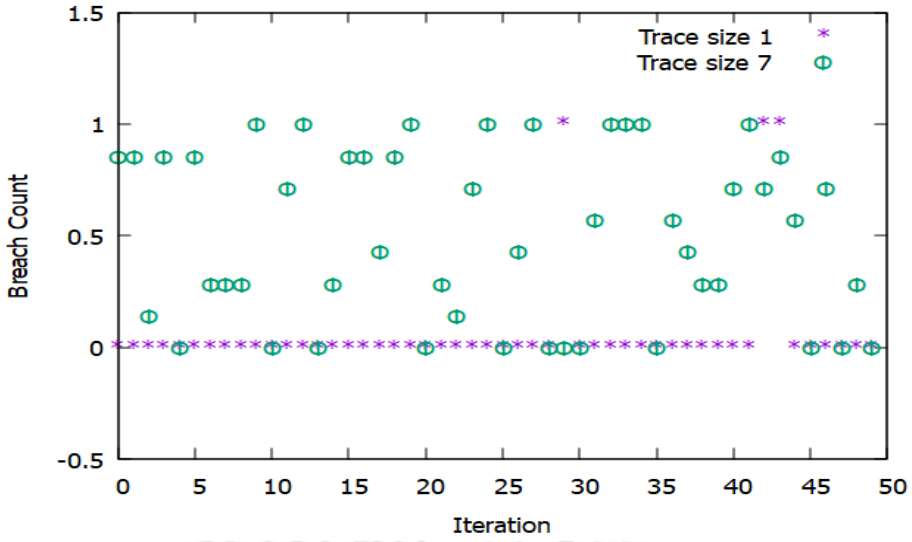
The first part of this section shows the results for fixed trace. In the second part, we vary the trace size within a range.

3.5.4.1 Fixed trace

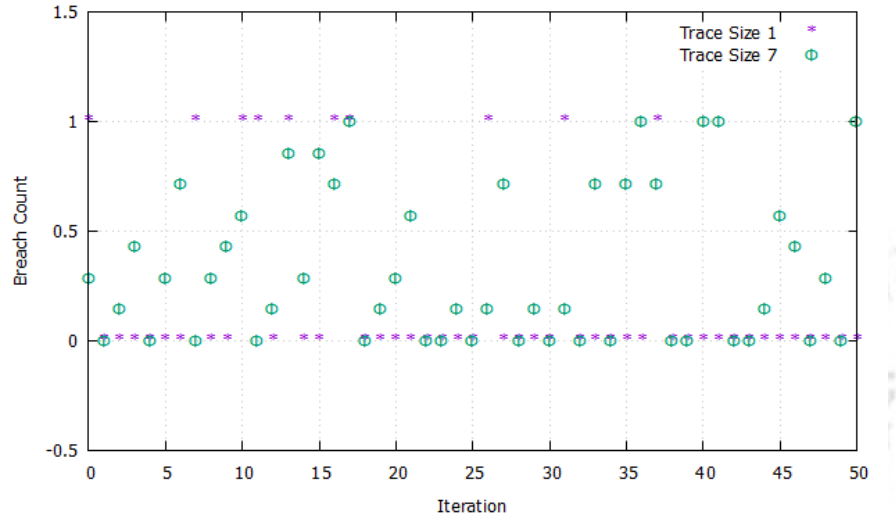
In this experiment, we first compare the privacy level achieved by a standard DP mechanism for a point query (trace size 1) and a continuous query of trace size 7. Since the theoretical privacy loss scales linearly with trace length, a moderate trace size of 7 was selected to illustrate the effect before varying the trace size in subsequent experiments. The standard DP mechanism used has been detailed in Section 3.2.4.

3.5.4.1.1 Comparison of Breach Count:

Figure 3.3a and Figure 3.3b show how the breach count varies with iterations for a standard DP mechanism and our proposed privacy framework, respectively. We will achieve the highest privacy when the breach count is 0 and the lowest when it is 1.



(a) With standard DP mechanism.



(b) With proposed mechanism.

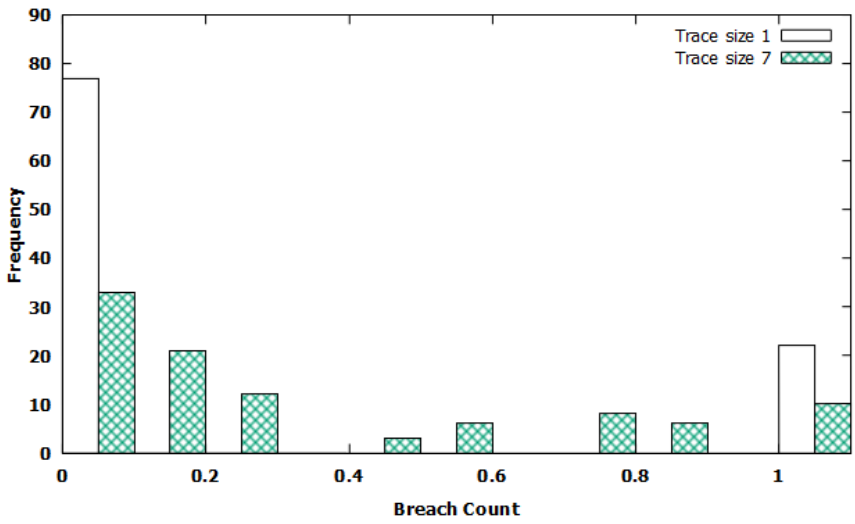
Figure 3.3: Comparison of Breach Count for trace size 1 and 7.

As can be seen from the figure, there are instances when both events occur. However, for trace size 1, a breach count of 0 is more frequent than a breach count of 1. For trace size 7, a breach count of 0 is less frequent. Further, we compare the privacy level achieved by the standard differential private (DP) mechanism with that of our proposed mechanism. The breach count values for trace size 7 are significantly lower for our proposed mechanism. In 50% of the cases, the proposed mechanism obtains a zero breach count whereas the same is 25% for the standard DP mechanism. This implies that our mechanism is better suited to handle continuous queries.

In Figure 3.4, we plot the distribution of the breach count values for trace sizes 1 and trace size 7 for the standard DP mechanism and our proposed mechanism, respectively. As seen from Figure 3.4a, the frequency of breach count being equal

to zero in case of point queries is almost 90%. For trace size 7, we get ideal privacy (i.e. breach count equal to zero) only in 22% of the cases. We also attain fractional values for breach count in the case of trace size 7. However, using our proposed mechanism we achieve a breach count of zero for trace size 7 more frequently.

(a) With standard DP mechanism.

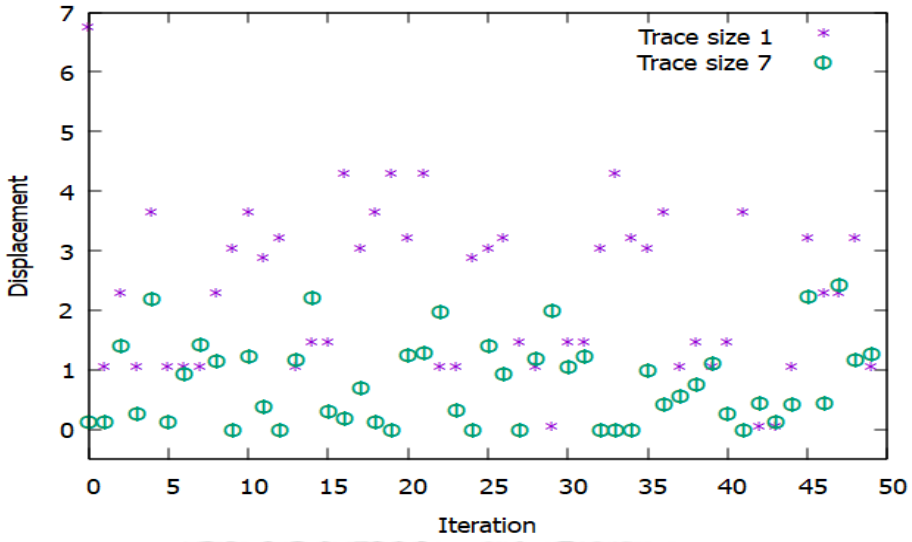


(b) With proposed mechanism.

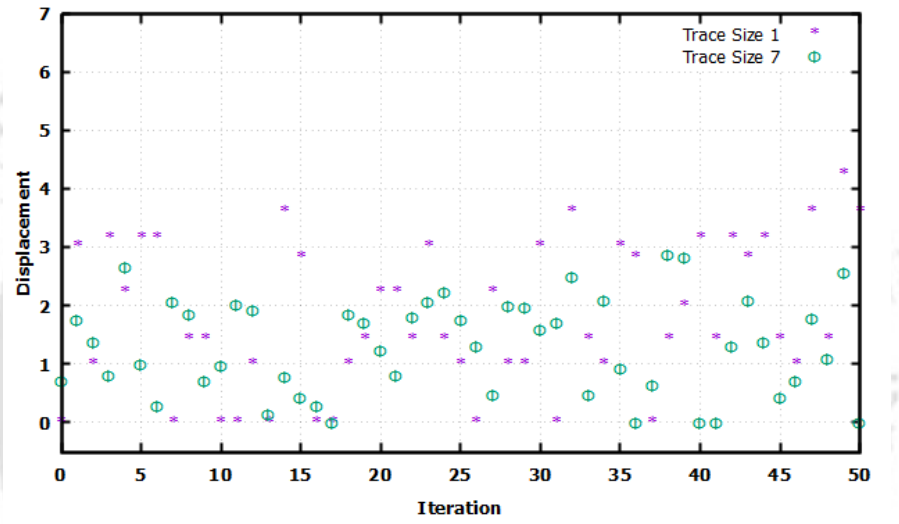
Figure 3.4: Breach Count Frequency for trace size 1 and trace size 7.

3.5.4.1.2 Comparison of Displacement:

Figure 3.5a and Figure 3.5b show the variation of displacement over 50 iterations for a standard differential (DP) mechanism and our proposed approach, respectively.



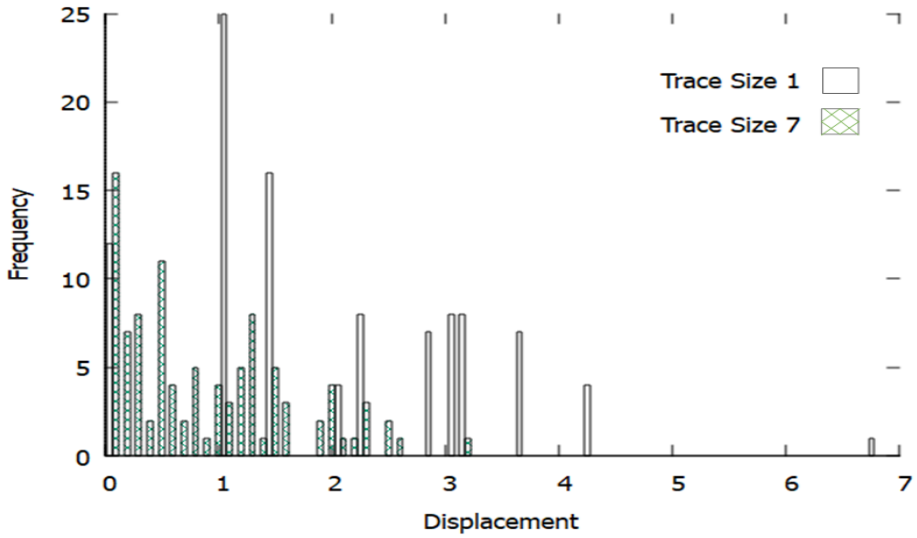
(a) With standard DP Mechanism.



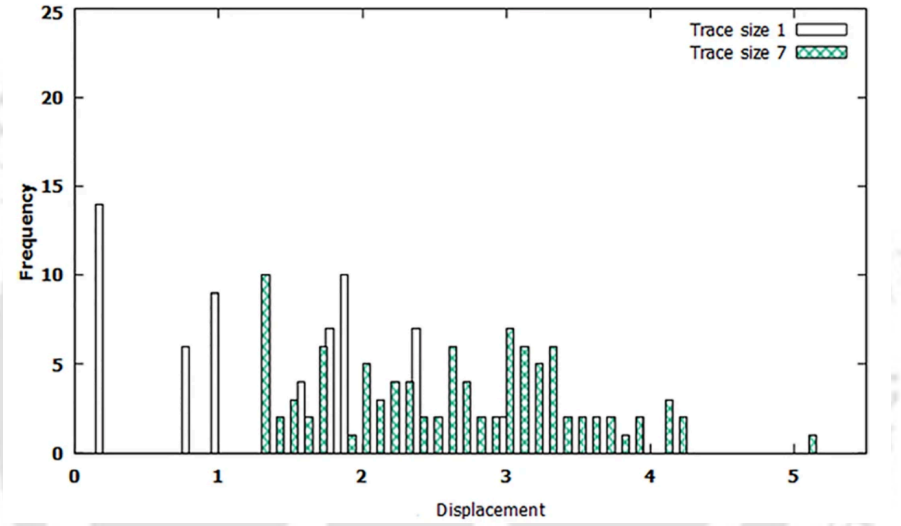
(b) With proposed mechanism.

Figure 3.5: Comparison of Displacement for trace size 1 and trace size 7.

Suppose a standard DP-preserving mechanism is used for point queries (Figure 3.5a). In that case, the average distance (averaged over a set of 100 results) between the actual and perturbed location(displacement) is 1.849 units. The same mechanism, when used on a trace of size 7, the average displacement drops to 0.944 units. That is, there is a significant drop of 50% in the displacement and privacy. The other conclusion that we can draw from this experiment is that the results also vary on the trace location from where the query is fired.



(a) With standard DP mechanism.



(b) With proposed mechanism.

Figure 3.6: Displacement Frequency for trace size 1 and trace size 7.

The average distance between the actual and perturbed location is lower for the standard DP mechanism as compared to that of our proposed mechanism. For trace size 7, the displacement values are clustered between 0 and 1 while using a standard DP mechanism (Figure 3.5a). On the other hand, for the proposed privacy mechanism, the displacement values are concentrated between 1 and 2 (Figure 3.5b). Higher displacement values between the actual and perturbed location imply a higher level of privacy. This indicates our proposed mechanism fares better in terms of privacy.

In Figure 3.6, we plot the distribution of the displacement values for trace size 1 and 7 for both privacy mechanisms. Expectedly, the proposed privacy framework achieves higher displacement values more frequently.

3.5.4.2 Progressive trace size

In this set of experiments, we examine how trace size affects the privacy outcome. We vary the trace size from 1 to 50.

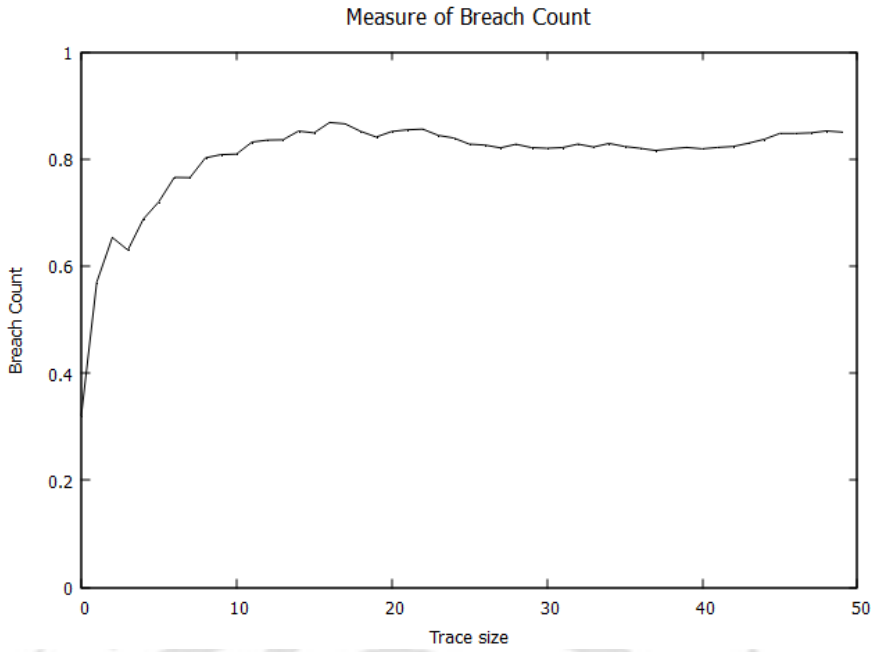
3.5.4.2.1 Variation of Breach Count and Displacement:

From Figure 3.7a it is clear that as trace size grows, the breach count also increases. That is, the privacy level decreases with trace size because the privacy level is inversely proportional to the breach count. The result is in concurrence with our theoretical analysis. It should be noted that breach count also depends on the location trace, hence in a few cases we find that the breach count value decreases with an increase in trace size. The result also shows that for smaller trace sizes (up to 20) the increment is rapid. It is interesting to note when the trace size increases from 1 to 2, there is almost a 40% increase in the breach count. This shows that a DP mechanism designed for point queries will not be effective for continuous queries.

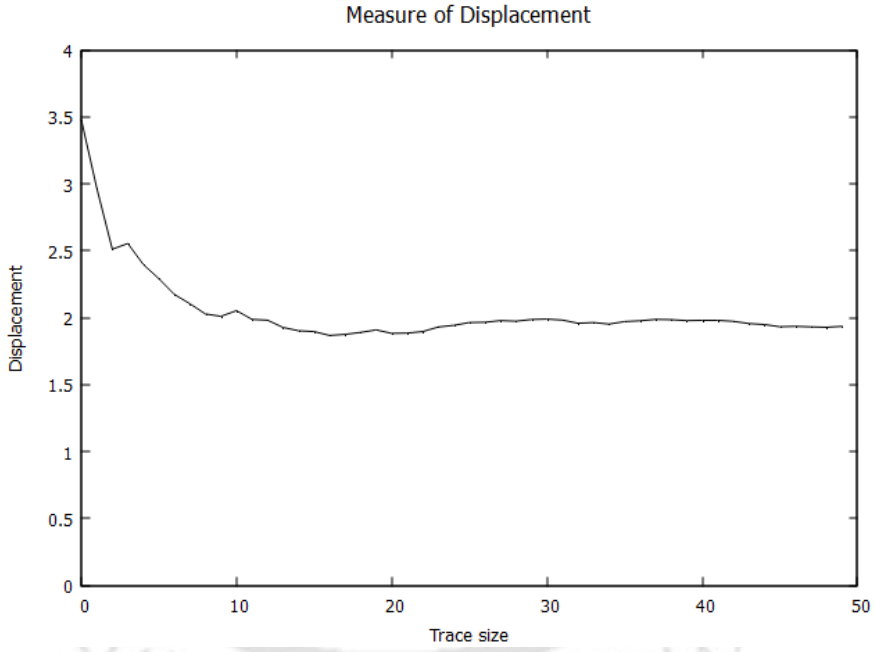
The other interesting point to note is that after a certain interval, the value of the breach count becomes almost constant. In our experiment, this constant value is about 83%. This observation indicates that, under the adopted experimental setting and attack model, increasing the trace size beyond this point does not substantially change the empirical breach count.

Figure 3.7b shows the correlation between trace size and displacement. It is clear that as the trace size increases the displacement decreases. That is to say, the privacy level decreases with trace size because the privacy level is directly proportional to displacement. It should be noted that displacement also depends on the location trace we use. Hence in some cases, the displacement value increases instead of decreasing. The result also shows that for smaller trace sizes (up to 20) the decrements are rapid, but after a point, it becomes almost constant.

This result is in agreement with that of breach count, which implies that after a certain value of trace size, the chosen attacker cannot significantly decrease the level of privacy.



(a) Breach count.



(b) Displacement.

Figure 3.7: Results for growing trace.

3.5.4.2.2 Variation of Drift Ratio:

In Figure 3.8, we plot the correspondence between the drift ratio and time stamp. As can be seen from the figure, the drift ratio is 1 for the first few time stamps, then decreases gradually. The result suggests that the true location is more likely to be included as we iterate. This means the privacy value will decrease as time evolves.

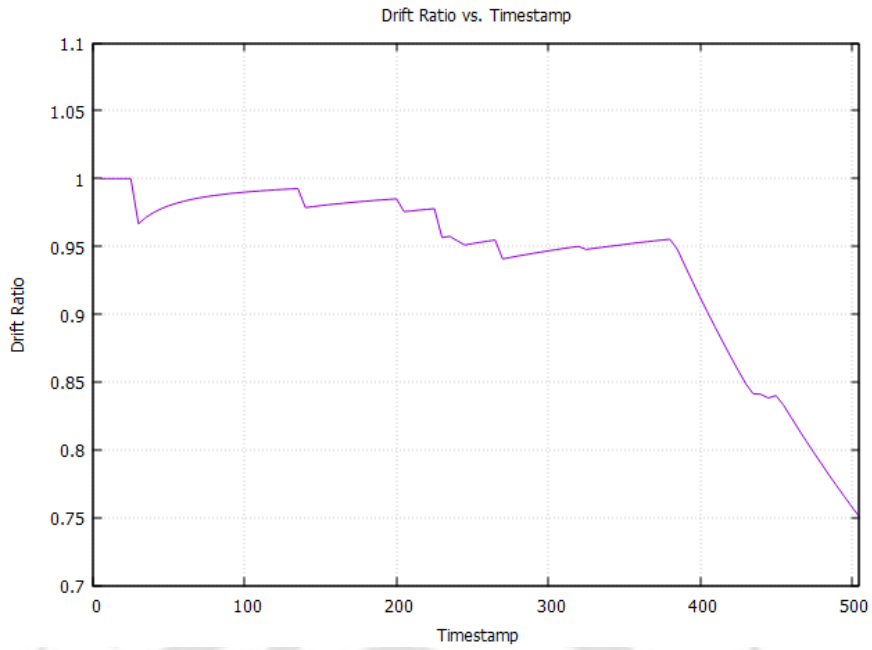


Figure 3.8: Drift ratio vs. time stamp.

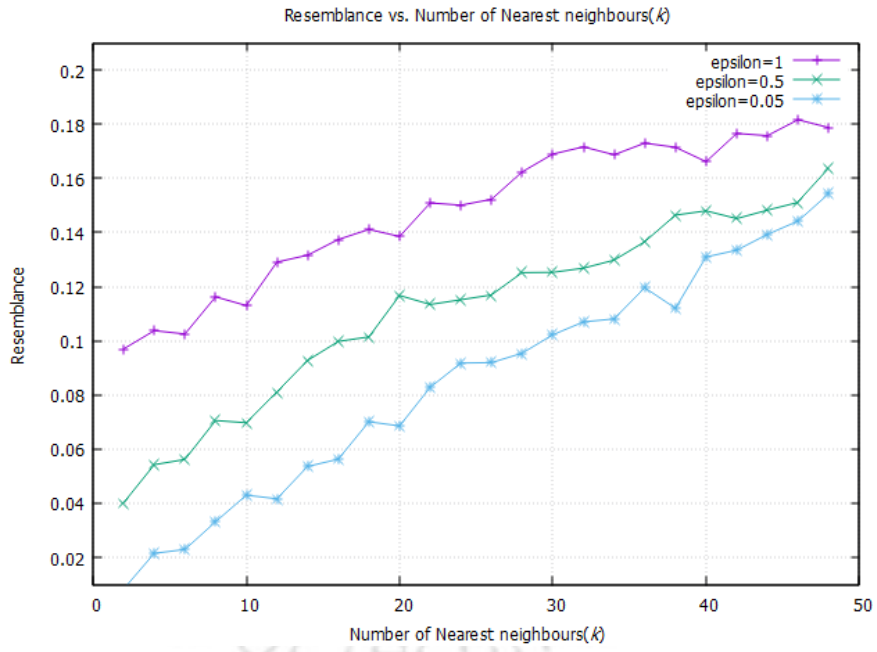


Figure 3.9: Resemblance vs. Number of Nearest neighbours(k).

3.5.4.2.3 Variation of Resemblance:

In Figure 3.9, we show the variation of resemblance with k nearest neighbours for three different values of privacy thresholds ϵ. It is evident from the figure that the resemblance increases with the increase in the value of k. Furthermore, higher values of ϵ yield higher resemblance values. On average, it has been seen that with ten units increment in the value of k, the resemblance values increase by about

18%. An increase in k will increase the search space for PoIs. As a result, overlap in the regions corresponding to the actual location and perturbed location will increase. This in turn increases the resemblance. Similarly, if we increase ϵ we are decreasing the level of privacy and as a result, the perturbation is also generated in a smaller region. Hence the actual and perturbed location will be closer which will increase resemblance.

3.6 Conclusion

In this chapter, we thoroughly audit DP for continuous queries. Although DP provides a provable privacy guarantee for point queries, the privacy level decreases under repetitive use on a set of correlated locations. The number of independent locations on which DP is applied is called the trace size. We detail how an adversary can exploit the mobility profile to compromise the privacy mechanism. We prove that for a given differential mechanism, if ϵ is the loss in privacy for point queries, then it is $n'\epsilon$, when applied to n' locations.

Experiments were performed using a number of performance parameters to corroborate the theoretical analysis. We proposed a performance metric breach count to empirically measure the loss of privacy. Our experiments show that by merely increasing the trace size to 2, there is more than a 40% decrease in the privacy level. The metric displacement quantifies the average distance between the actual and perturbed locations. Using a standard DP measure, we find that the displacement value decreases by 50%. We also got similar results for the other two privacy metrics, drift ratio and resemblance. These results indicate that for continuous queries a user's actual location is more likely to be compromised by considering past movement, mobility profile and topological constraints.

We also proposed a perturbation based privacy framework for continuous queries. The proposed mechanism generates the perturbed location by using the neighbouring locations and filtering out the ones which do not satisfy the utility requirement of the user. If we fail to find a feasible perturbed location which satisfies the utility requirement, we resort to the standard noise addition mechanism, thereby guaranteeing privacy preservation in all scenarios. Experimental results show our proposed mechanism improves the privacy level compared to a standard DP measure.

Chapter 4

Utility Aware Adaptive Differential Privacy Budget Allocation in a Streaming MAS

This chapter addresses privacy-utility tradeoffs in MAS, where fixed DP budget may either obscure critical events or exhaust the budget prematurely. A dynamic method is proposed that allocates privacy budget depending on the usefulness of an agent's observation and the extent to which the data influences the overall system estimate [1].

[1]The contents of this chapter appears in the paper titled "Utility Aware Adaptive Privacy Budget Allocation for Streaming Multi-Agent Systems", in the Proceedings of the 25th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2026: CORE A* Conference).

In the previous chapter, we saw that standard differential privacy (DP) with uniform per timestep privacy budget allocation fails for mobile agents which fire continuous queries. We saw that ensuring per timestep ϵ-DP leads to a degradation of privacy over multiple timesteps. Temporally evolving data streams are common in MASs and inadvertently leak sensitive and strategic information [4] about the agent, such as their behavioral patterns [66]. In this chapter, we extend it to cases where the shared data contains more than just location information. DP budgets are finite and accumulate under sequential composition. To maintain an ϵ-DP guarantee over a total of t^* timesteps, it can be distributed uniformly to provide $\frac{\epsilon}{t^*}$-DP per timestep. Such uniform distribution implies enhanced privacy per timestep due to a large magnitude of added noise, and consequently, degraded utility. We further illustrate the issues with an example.

In case of the smart grid network, if too much noise is added uniformly across all reports, the utility provider will miss critical load surges or anomalies, degrading system reliability. Conversely, too little noise results in privacy risks getting compounded over time as temporal correlations can be exploited to reconstruct sensitive usage patterns. This example illustrates that, for streaming MASs, if differential privacy mechanisms are used, they must adapt to signal dynamics to avoid wasting budget in uninformative periods and thereby spending appropriately in high impact ones.

To summarize, an uniformly allocated budget and the subsequent addition of a fixed level of noise per timestep leads to two key problems:

1. Utility degradation: Informative timesteps with high variance are overly perturbed which affects tasks like forecasting or anomaly detection.

2. Inefficient budget use: Budgets are consumed at the same rate regardless of informativeness, leading to premature budget exhaustion or wasted budget during quiescent periods.

Thus, streaming MASs require privacy mechanisms that adaptively allocate noise over time, ensuring that privacy budgets are spent where they yield the highest utility. In this work, we propose adaptive privacy budget allocation (APBA), which is a mechanism that dynamically distributes each agent's finite DP budget across timesteps. This allocation is guided by both the informativeness of the agent's observations and the potential impact of its data on the overall estimate of the system. Unlike fixed noise [35] or clipping based [6] approaches, APBA allocates more budget (less noise) during volatile periods and less budget (more noise) during stable periods, resulting in a principled privacy-utility tradeoff.

APBA is explicitly designed for analytics first scenarios: its goal is to maximize estimation accuracy subject to a global privacy guarantee, not to maximize privacy

at all costs. This design leads to a controlled but higher reconstruction similarity compared to conservative baselines, an intentional choice to preserve critical signal fidelity.

4.1 System Model

We consider the MAS model illustrated in Figure 1.1 in which S autonomous agents send their observations about a system parameter $\mathbf{x}$ at regular intervals to a central fusion center. At discrete timesteps $t \in \{1, \ldots, t^*\}$, each agent i observes a vector $\mathbf{y}_i(t) \in \mathbb{R}^n$ representing its local measurement. The structure of the observation tuple of the i-th agent at the t-th step is as follows:

$$\mathbf{y}_i(t) = (p_{1t}, p_{2t}, \ldots, p_{kt}, g_{1t}, \ldots, g_{(n-k)t}) \in \mathbb{R}^n. \tag{4.1}$$

The tuple elements $\{p_{jt}\}_{j=1}^{k}$ are the private parameters while the rest are public observation parameters.

These observations are temporally correlated, reflecting the system dynamics or environment being monitored. Agents communicate their measurements to a central fusion center, which produces an estimate of an aggregated system parameter $\hat{\mathbf{x}}(t)$, which is a statistic derived across all agents.

The following attributes are inherent to each agent i:

- Total privacy budget: $\epsilon_i^{\text{total}}$, representing the maximum allowable cumulative privacy loss over the streaming horizon.
- Remaining privacy budget: $\epsilon_i^{\text{remaining}}(t)$, which decreases over time as the agent perturbs observations.
- Local uncertainty measure: $u_i(t)$, computed using recent measurements to quantify variability or unpredictability. Suitable volatility metrics include variance and interquartile range among a few.

At each timestep t, an agent perturbs its observation $\mathbf{y}_i(t)$ using a noise addition based mechanism (as introduced in Chapter 3) with an adaptive privacy budget $\epsilon_i(t)$. We replace Δq with $\Delta \mathbf{y}$, which denotes the sensitivity of the observation, that is, the maximum possible change in $\mathbf{y}_i(t)$ between two consecutive observations.

The magnitude of noise added is inversely proportional to the privacy budget $\epsilon_i(t)$. The adaptive allocation $\epsilon_i(t)$ is computed dynamically. Higher privacy budget $\epsilon_i(t)$ would imply less noise, while a lower value implies higher magnitude of added noise, and thereby higher levels of privacy.

The fusion center aggregates the perturbed reports $\{\tilde{\mathbf{y}}_i(t)\}_{i=1}^{S}$ to compute an estimate of the system parameter $\hat{\mathbf{x}}(t)$. It continuously updates $\hat{\mathbf{x}}(t)$ as new observations arrive. The adaptive privacy allocation directly affects estimation quality and enables the fusion center to receive informative reports while agents' privacy constraints are maintained.

The system operates continuously, with both agent observations $\mathbf{y}_i(t)$ and adaptive privacy allocations $\epsilon_i(t)$ evolving over time. By exploiting temporal correlations, agents adjust noise magnitude according to signal volatility, ensuring the fusion center receives informative reports when needed. This dynamic allocation balances privacy and estimation utility, outperforming static, fixed noise mechanisms.

Each agent computes $\epsilon_i(t)$ and the fusion center aggregates these reports to estimate $\hat{\mathbf{x}}(t)$. This distributed coordination is central to streaming MAS applications.

4.2 Methodology

The proposed approach, APBA, is a streaming friendly mechanism that lets agents intelligently distribute differential privacy budgets over time while maximizing system utility. The proposed framework consists of three main components: (i) adaptive privacy budget allocation per agent, (ii) local perturbation of observations, and (iii) fusion center aggregation. The proposed methodology is briefly illustrated in Figure 4.1.

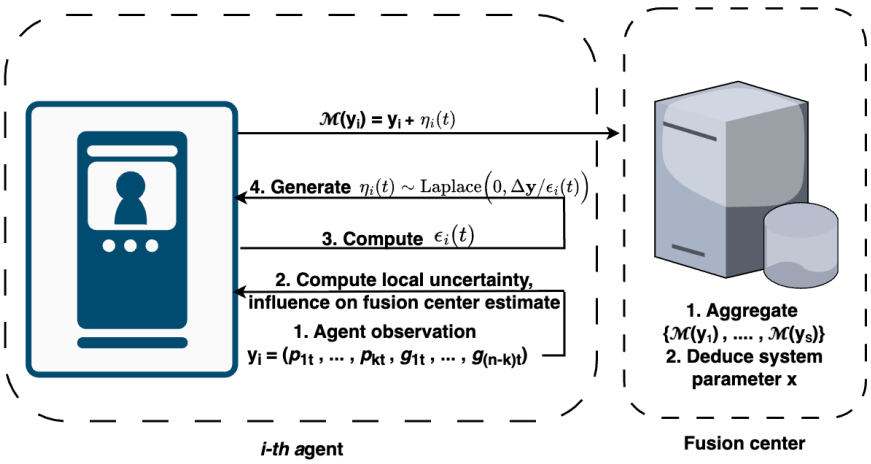


Figure 4.1: Proposed Dynamic Budget Allocation and Sanitization Methodology.

4.2.1 Adaptive privacy budget allocation

At each timestep t, agent i allocates a fraction of its remaining privacy budget, $\epsilon_i(t)$, based on two main factors:

1. Local signal uncertainty $u_i(t)$: Higher variance or volatility in $\mathbf{y}_i(t)$ indicates that the observation carries more information. To preserve utility, more budget (less noise) is allocated when uncertainty is high.

2. Influence on fusion center estimate $\Delta_i(t)$: Defined as

$$\Delta_i(t) = |\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{-i}(t)|,$$

where $\hat{\mathbf{x}}_{-i}(t)$ is the aggregated estimate without agent i. This quantifies the agent's contribution to the overall system estimate and guides allocation to maximize utility. Larger influence motivates higher budget allocation to ensure critical contributions are accurately captured. In practice, $\Delta_i(t)$ can be computed at the fusion center or estimated locally using historical reports.

The adaptive budget is computed as a normalized combination:

$$\epsilon_i(t) = \epsilon_i^{\text{remaining}}(t) \cdot \frac{\alpha u_i(t) + (1-\alpha)\Delta_i(t)}{\sum_{j=1}^{S} \left(\alpha u_j(t) + (1-\alpha)\Delta_j(t)\right)},$$

where $\alpha \in [0, 1]$ balances the relative importance of local uncertainty versus global influence. APBA allocates the per-timestep privacy budget based on an estimate of information importance derived from signal dynamics. Specifically, higher privacy budgets are assigned to timesteps where either (i) the local signal uncertainty $u_i(t)$ is large, or (ii) the agent's contribution to the global estimate, measured by $\Delta_i(t)$, is significant. Intuitively, these correspond to periods where accurate reporting is most valuable for downstream estimation.

Additionally, the total budget constraint is enforced:

$$\sum_{\tau=1}^{t} \epsilon_i(\tau) \leq \epsilon_i^{\text{total}}.$$

The remaining privacy budget for agent i evolves according to:

$$\epsilon_i^{\text{remaining}}(t) = \epsilon_i^{\text{total}} - \sum_{\tau=1}^{t} \epsilon_i(\tau).$$

The adaptive budget allocation variables are used as control parameters for selecting the per-timestep budget. The formal DP analysis presented in this chapter

concerns the sequence of released sanitized observations generated after perturbation. The intermediate quantities are low-dimensional summaries used during budget allocation, including the uncertainty and influence measures, and are not considered part of the final released outputs. Each agent computes its local uncertainty measure, while the normalization and influence computation are performed within the fusion-center protocol to determine the adaptive allocation. Only the sanitized observations are considered released outputs in the privacy analysis; the intermediate allocation variables are not part of the published data. Accordingly, the differential privacy guarantee proved in this chapter applies to the released sanitized reports and not to the auxiliary allocation variables.

4.2.2 Local perturbation of observations

Once $\epsilon_i(t)$ is determined, agent i perturbs its observation using the Laplace mechanism:

$$\tilde{\mathbf{y}}_i(t) = \mathbf{y}_i(t) + \eta_i(t), \quad \eta_i(t) \sim \text{Laplace}\Big(0, \frac{\Delta \mathbf{y}}{\epsilon_i(t)}\Big).$$

By sequential composition, the total privacy loss over t^* timesteps satisfies:

$$\sum_{t=1}^{t^*} \epsilon_i(t) \leq \epsilon_i^{\text{total}},$$

ensuring that each agent adheres to its allocated ϵ-DP budget. The adaptive allocation concentrates budget on informative or high impact timesteps, improving utility while preserving formal privacy guarantees.

When $\epsilon_i^{\text{remaining}}(t) = 0$, the agent can no longer produce fully differentially private observations. Once the agent's privacy budget is exhausted, one of the following approaches can be undertaken:

- Coarsened Reporting: Agents may provide aggregated temporal statistics requiring no additional budget.
- Budget Redistribution: Unused budgets from other agents can be reallocated to critical agents.
- Noise Masking: Agents may submit pure noise to maintain privacy while minimally contributing to system parameter estimation.

The Algorithm 3 outlines our proposed method. At each timestep, the algorithm computes both the local uncertainty of the agent's observation and its impact on the fusion center estimate. These factors are used to adaptively allocate a portion of the remaining privacy budget. Agents with available budget release perturbed observations with noise calibrated to the allocated privacy level,

Algorithm 3 APBA for Streaming Multi-Agent Systems

Require: Number of agents S, total timesteps t^*, privacy budgets $\{\epsilon_i^{\text{total}}\}$, observations $\{\mathbf{y}_i(t)\}$
Ensure: Streaming estimates $\{\hat{\mathbf{x}}(t)\}$ and perturbed reports $\{\tilde{\mathbf{y}}_i(t)\}$

```
Initialize ε_i^remaining ← ε_i^total for all agents
for t = 1 to t* do
    Compute local uncertainties u_i(t) for all agents
    Compute fusion center impact Δ_i(t) for all agents
    for each agent i do
        if ε_i^remaining > 0 then
            Allocate ε_i(t)
            Generate perturbed report ỹ_i(t) = y_i(t) + η_i(t)
            Update remaining budget: ε_i^remaining ← ε_i^remaining − ε_i(t)
        else
            Apply coarsened reporting
            Set ỹ_i(t) accordingly
        end if
    end for
    Fusion center computes: x̂(t)
end for
return {x̂(t)}, {ỹ_i(t)}
```

while updating their remaining budget accordingly. Once the budget is exhausted, agents switch to coarsened reporting that incurs no additional privacy loss. The fusion center aggregates all received reports and computes the updated global estimate. The decentralized budget allocation has $\mathcal{O}(1)$ computation per agent per timestep, since each agent updates its budget using only local statistics (e.g., running variance and remaining budget).

Both budget allocation and fusion center aggregation adapt dynamically over time. Agents assign larger budgets during periods of high volatility and smaller budgets during stable periods, achieving a principled tradeoff between privacy and system utility across the streaming horizon.

4.3 Theoretical Guarantees

This section provides formal guarantees for the proposed APBA mechanism. In Theorem 4.1, we show that APBA guarantees $(\epsilon_i^{\text{total}}, 0)$-differential privacy even under adaptive allocations. The guarantee continues to hold despite the gradual depletion of the privacy budget, while also minimizing estimation variance relative to uniform allocation. In Corollary 4.3, we show that the proposed approach preserves unbiasedness. Furthermore, the aggregated estimate converges to the true parameter as S increases asymptotically, provided the estimation function at the fusion center is designed appropriately. The mechanism further degrades

gracefully under partial agent dropout and achieves a strictly better privacy–utility tradeoff for streaming data characterized by high variance.

4.3.1 Differential Privacy Guarantees

To demonstrate the DP guarantee, we first show in Lemma 4.1 that the cumulative privacy loss over t^* timesteps is the sum of the individual per-step budgets. The results hold even when the per-step budgets vary over time.

Lemma 4.1 (Sequential Composition under Adaptive Allocation) *Let $\mathcal{M}_i(t)$ denote the mechanism applied by agent i in the timestep t with budget $\epsilon_i(t)$. Then the cumulative mechanism over t^* timesteps satisfies*

$$\mathcal{M}_i^{1:t^*} \text{ is } \Big(\sum_{t=1}^{t^*} \epsilon_i(t), 0\Big)\text{-DP.}$$

Proof. Each mechanism $\mathcal{M}_i(t)$ is $(\epsilon_i(t), 0)$-DP by construction. By the standard sequential composition theorem [23], applying these mechanisms over t^* timesteps results in cumulative privacy loss equal to the sum of their individual budgets. Hence, $\mathcal{M}_i^{1:t^*}$ is $\big(\sum_{t=1}^{t^*} \epsilon_i(t), 0\big)$-DP. □

When $\epsilon_i^{\text{remaining}}(t) = 0$, APBA switches the agent i to coarse reporting. Let $\epsilon_i^{\text{coarse}}$ denote the (negligible) privacy cost of this fallback mechanism. The cumulative privacy loss is therefore the main concern.

$$\epsilon_i^{\text{cumulative}}(t^*) = \sum_{t=1}^{t^*} \epsilon_i(t) + \epsilon_i^{\text{coarse}} \leq \epsilon_i^{\text{total}}.$$

Theorem 4.1 (Privacy Guarantee with Budget Exhaustion) *If each agent follows APBA with total budget ϵ_i^{total}, then APBA guarantees $(\epsilon_i^{total}, 0)$-differential privacy for the entire streaming horizon $t = 1, \ldots, t^*$, including after budget exhaustion.*

Proof. Lemma 4.1 ensures that the cumulative privacy loss is $\sum_{t=1}^{t^*} \epsilon_i(t)$. By construction, APBA enforces $\sum_{t=1}^{t^*} \epsilon_i(t) \leq \epsilon_i^{\text{total}} - \epsilon_i^{\text{coarse}}$, and $\epsilon_i^{\text{coarse}} \ll \epsilon_i^{\text{total}}$. Hence, the total loss of privacy never exceeds $\epsilon_i^{\text{total}}$, which satisfies the global privacy guarantee. □

This formally establishes that each agent's cumulative privacy loss is bounded over the entire streaming horizon, ensuring no additional privacy loss occurs beyond budget exhaustion.

4.3.2 Utility Guarantees

We establish a bound on the estimation error incurred due to Laplace noise addition under APBA mechanism. Theorem 4.2 quantifies how the local noise injections propagate through the fusion center estimator. The mean squared estimation error is bounded.

Theorem 4.2 (Bounded Estimation Error) *Let $\tilde{\mathbf{y}}(t) = (\tilde{\mathbf{y}}_1(t), \ldots, \tilde{\mathbf{y}}_S(t))$ be the vector of reports from noisy agents. Let the fusion center estimator be any deterministic map $\hat{\mathbf{x}}(t) = f(\tilde{\mathbf{y}}(t))$, where $f : \mathbb{R}^S \to \mathbb{R}$ is $\beta(\geq 0)$-Lipschitz with respect to the Euclidean norm:*

$$|f(u) - f(v)| \leq \beta \|u - v\|_2 \quad \forall\, u, v \in \mathbb{R}^N.$$

Then the mean squared estimation error satisfies

$$\mathbb{E}\big[\big|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\big|^2\big] \leq 2\beta^2 \sum_{i=1}^{S} \frac{(\Delta \mathbf{y})^2}{\big(\epsilon_i(t) + \epsilon_i^{\text{coarse}}\big)^2},$$

where $\mathbf{x}(t) = f(\mathbf{y}(t))$ is the estimator applied to the noiseless inputs.

Proof. Let $\eta = (\eta_1, \ldots, \eta_S)$. By definition, $\hat{\mathbf{x}}(t) = f(\mathbf{y}(t) + \eta)$ and $\mathbf{x}(t) = f(\mathbf{y}(t))$. By the Lipschitz property of f,

$$|\hat{\mathbf{x}}(t) - \mathbf{x}(t)|^2 = |f(\mathbf{y} + \eta) - f(\mathbf{y})|^2 \leq \beta^2 \|\eta\|_2^2.$$

Taking expectations gives

$$\mathbb{E}\big[|\hat{\mathbf{x}}(t) - \mathbf{x}(t)|^2\big] \leq \beta^2 \mathbb{E}\big[\|\eta\|_2^2\big] = \beta^2 \sum_{i=1}^{S} \mathbb{E}[\eta_i^2].$$

For $b_i = \frac{\Delta \mathbf{y}}{\epsilon_i(t) + \epsilon_i^{\text{coarse}}}$ and $\eta_i \sim \text{Laplace}(0, b_i)$ we have $\text{Var}(\eta_i) = 2b_i^2$, so

$$\mathbb{E}[\eta_i^2] = 2\left(\frac{\Delta \mathbf{y}}{\epsilon_i(t) + \epsilon_i^{\text{coarse}}}\right)^2.$$

Substituting into the previous expression yields

$$\mathbb{E}\big[|\hat{\mathbf{x}}(t) - \mathbf{x}(t)|^2\big] \leq 2\beta^2 \sum_{i=1}^{S} \frac{(\Delta \mathbf{y})^2}{\big(\epsilon_i(t) + \epsilon_i^{\text{coarse}}\big)^2},$$

which proves the claim. □

In Corollary 4.3, we prove that as the number of participating agents becomes

sufficient, the noise effect diminishes at the rate of $\frac{1}{S}$. Thus the expected estimate converges to the true parameter.

Corollary 4.3 (Convergence of Streaming Estimates) *If $\mathbf{y}_i(t)$ are bounded and $S \to \infty$, then*

$$\lim_{S\to\infty} \mathbb{E}[\hat{\mathbf{x}}(t)] = \mathbf{x}(t),$$

up to noise variance, which vanishes at rate $1/S$ when $\beta = \mathcal{O}(1/S)$. Thus, APBA preserves asymptotic unbiasedness of the aggregated estimate.

These results show that APBA not only controls the error induced by adaptive noise injection but also guarantees convergence of the aggregated estimate to the true parameter as the number of participating agents increases, providing a formal foundation for the mechanism's reliability and applicability to large-scale MAS.

4.3.3 Robustness to Dropout

We now extend the error analysis to account for agents that have exhausted their privacy budgets and thereafter switched to coarse reporting. The following corollary proves that even under such a scenario, the mean squared estimation error remains bounded.

Corollary 4.4 (Bounded Error under Agent Dropout) *Let $\mathcal{D} \subseteq \{1, \ldots, S\}$ be the subset of agents who have exhausted their privacy budgets and switch to coarse reporting. Let $\hat{\mathbf{x}}(t) = f(\tilde{\mathbf{y}}(t))$ be an unbiased β-Lipschitz estimator of $\mathbf{x}(t) = f(\mathbf{y}(t))$. Then the mean squared error satisfies*

$$\mathbb{E}\Big[\big|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\big|^2\Big] \;\le\; 2\beta^2 \sum_{i\in\{1,\ldots,S\}\setminus\mathcal{D}} \frac{(\Delta \mathbf{y})^2}{\epsilon_i(t)^2} \;+\; \mathrm{Var}_{\mathrm{coarse}}(\mathcal{D}),$$

where $\mathrm{Var}_{\mathrm{coarse}}(\mathcal{D})$ denotes the total contribution to the error of the coarsened (fallback) reports of agents in $\mathcal{D}$.

Thus, the estimator exhibits graceful degradation as agents drop out, maintaining a finite error bound even when some agents use coarse reporting.

4.3.4 Adaptive Variance Reduction

To theoretically prove the effectiveness of APBA, we analyze its privacy utility behaviour with respect to uniform budget allocation. In particular, we establish formal guarantees on estimation variation, mean squared error (MSE), and optimal budget allocation under a fixed privacy constraint. By adaptively directing the budgets, APBA reduces the noise contribution for informative timesteps. The

following lemma formalizes this variance reduction relative to uniform per timestep allocation.

Lemma 4.2 (Variance Reduction via APBA) *Let* $Var_{APBA}[\hat{\mathbf{x}}(t)]$ *and* $Var_{Uniform}[\hat{\mathbf{x}}(t)]$ *be the variances under APBA and uniform allocation, respectively. Then*

$$Var_{APBA}[\hat{\mathbf{x}}(t)] \leq Var_{Uniform}[\hat{\mathbf{x}}(t)] \quad \textit{for high uncertainty timesteps.}$$

Proof. APBA assigns higher $\epsilon_i(t)$ to timesteps with high $u_i(t)$ or $\Delta_i(t)$, reducing added noise precisely when the signal variability is highest. Uniform allocation cannot exploit this temporal heterogeneity, resulting in strictly higher variance during such periods. □

4.3.5 Privacy–utility improvement

Lemma 4.3 (Improved Privacy-Utility Tradeoff) *Let the total privacy budget of the i-th agent be* ϵ_i^{total} *over* t^* *timesteps. Define the effective weighted mean squared error as a heuristic bound:*

$$\text{MSE}_i := \sum_{t=1}^{t^*} \frac{w_t}{(\epsilon_i(t) + \epsilon_i^{\text{coarse}})^2}, \quad w_t \equiv w_i(t) := u_i(t)^2 + \Delta_i(t)^2,$$

with w_t *being the timestep importance weight. Then the allocation that minimizes* MSE_i *under the budget constraint*

$$\sum_{t=1}^{t^*} \epsilon_i(t) \leq \epsilon_i^{total}, \quad \epsilon_i(t) \geq 0,$$

satisfies

$$\epsilon_i(t) + \epsilon_i^{\text{coarse}} \propto w_t^{1/3}.$$

Consequently, the APBA method reduces the expected error compared to uniform allocation in non-stationary settings.

Proof. In streaming MASs, some timesteps contribute more to the MSE due to larger signal uncertainty or larger influence on the fusion center. We want to minimize agent-specific MSE under the privacy budget, that is,

$$\min_{\epsilon_i(t)\geq 0} \sum_{t=1}^{t^*} \frac{w_t}{\left(\epsilon_i(t) + \epsilon_i^{\text{coarse}}\right)^2}, \quad \text{s.t.} \ \sum_{t=1}^{t^*} \epsilon_i(t) \leq \epsilon_i^{\text{total}}.$$

This is convex in $\epsilon_i(t)$. Introducing a Lagrange multiplier $\mu \geq 0$ and by subsequently setting the partial derivative with respect to $\epsilon_i(t)$ to zero, we have

$$-\frac{2w_t}{\left(\epsilon_i(t) + \epsilon_i^{\text{coarse}}\right)^3} + \mu = 0$$
$$\implies \quad \epsilon_i(t) + \epsilon_i^{\text{coarse}} = \left(\frac{2w_t}{\mu}\right)^{1/3}.$$

The optimal allocation is cubic-root weighted in w_t. APBA preserves this monotonic structure by assigning larger budgets to larger w_t, thereby moving toward the optimal allocation and reducing the weighted MSE relative to uniform allocation in heterogeneous streaming environments. APBA's adaptive rule therefore achieves an improved privacy-utility trade-off over uniform allocation which distributes the budget equally across all timesteps and therefore adds relatively more noise to high-impact steps, resulting in higher overall estimation error. Consequently, for the same privacy loss, APBA improves utility relative to uniform allocation, yielding a better privacy-utility tradeoff in non-stationary streaming settings. □

Although Theorem 4.2 provides a uniform upper bound, weighting timesteps reflects their true contribution to the expected squared error. This weighted MSE is convex and thus tractable for optimization, guiding APBA to allocate more budget where it matters most.

The update follows from minimizing a convex inverse-square error proxy under a linear global privacy constraint via Lagrangian relaxation. The resulting allocation scales as a power of the timestep importance, ensuring nonnegativity and satisfaction of the total privacy budget constraint.

4.4 Experimentation

We evaluate our proposed APBA mechanism on multiple streaming multi-agent datasets, comparing it against state of the art baselines in terms of both utility and privacy leakage.

4.4.1 Datasets

We used two streaming datasets with distinct temporal characteristics:

1. **Intel Lab Sensor Dataset [8]:** consists of measurements from 10 spatially distributed sensors over 1000 timesteps. We use the temperature and humidity streams collected in an indoor laboratory environment. The data exhibits

high-frequency fluctuations and occasional abrupt spikes caused by environmental changes and sensor noise, resulting in short-term nonstationarity. These characteristics make the dataset representative of streaming sensing applications in which local uncertainty varies significantly over time.

2. **NREL Wind Turbine Dataset [25]:** contains wind energy generation profiles from 8 geographically distributed sites over 1000 timesteps. Unlike the Intel dataset, the NREL data follows smooth diurnal and meteorological patterns with moderate day-to-day variability, leading to relatively structured and quasi-stationary temporal dynamics. This dataset enables evaluation of APBA under lower volatility and more predictable evolution, thereby highlighting how adaptive allocation behaves in comparatively stable environments.

These two datasets allow us to evaluate APBA in both highly volatile and smoother temporal settings.

4.4.2 Baselines

The proposed approach is compared with two baseline approaches.

- **Shuffled DP (S-DP)** [42]: implements a two-stage privacy mechanism in which each user first applies local randomization to their data and then the tuple elements are randomly shuffled before aggregation. The shuffling step provides privacy amplification, allowing the system to achieve stronger central privacy guarantees than pure local differential privacy at comparable noise levels. In this setting, the local privacy parameter local ϵ quantifies the privacy protection of each user's randomized message prior to shuffling, while the central privacy parameter central ϵ characterizes the overall privacy guarantee after aggregation. Shuffled DP thus serves as a strong baseline that uses uniform distribution of privacy budget and reduces the noise through anonymity rather than through adaptive budget allocation.

- **FedAPCA (F-APCA)** [72]: is a hierarchical federated learning framework that incorporates adaptive local differential privacy. Clients are organized in a multi-level aggregation structure, and privacy budgets are adjusted across clients to improve the privacy–utility tradeoff. Unlike fixed-budget local DP schemes, FedAPCA dynamically distributes privacy budgets based on client contributions while maintaining a global central differential privacy constraint. Its primary control parameter is the central privacy budget ϵ, which governs the overall privacy guarantee of the federated aggregation process.

This baseline is particularly relevant because it incorporates adaptive allocation, though it does not explicitly model temporal volatility in streaming settings.

- **Variance Weighted (VW)**: scheme allocates privacy budgets proportionally to the empirical variance of each agent's data stream, i.e., $\epsilon_i(t) \propto \mathrm{Var}_i(t)$. The intuition is that agents exhibiting higher variability contribute more uncertainty to the global estimate and therefore receive larger privacy budgets (less noise). This approach is grounded in classical distributed estimation and weighted consensus principles [11], where weights are assigned based on signal quality or variance to improve estimation accuracy. However, VW relies solely on instantaneous variance statistics and does not explicitly account for temporal influence or long-term budget constraints.

4.4.3 Adversarial Reconstruction Model

To evaluate the robustness of the proposed APBA mechanism, we consider a strong third-party adversary performing a reconstruction attack on reported noisy data streams. The adversary is assumed to have complete knowledge of the aggregation protocol, the noise distribution, and the global statistics of the system.

The adversary seeks to reconstruct the original signals $\{\mathbf{y}_i(t)\}$ corresponding to the perturbed signal $\{\tilde{\mathbf{y}}_i(t)\}$ of agent i. This is done by solving a convex optimization problem that balances fidelity to observed reports with temporal smoothness constraints. Specifically, the adversary solves the following:

$$\min_{\{\hat{\mathbf{y}}_i(t)\}} \sum_{i=1}^{S} \sum_{t=1}^{t^*} \left(\hat{\mathbf{y}}_i(t) - \tilde{\mathbf{y}}_i(t)\right)^2 + \lambda \sum_{i=1}^{S} \sum_{t=2}^{t^*} \left(\hat{\mathbf{y}}_i(t) - \hat{\mathbf{y}}_i(t-1)\right)^2,$$

where $\hat{\mathbf{y}}_i(t)$ denotes the reconstructed signal and $\lambda > 0$ is a regularization parameter controlling the temporal smoothness. The first term enforces proximity to the noisy observations, while the second term penalizes rapid fluctuations to exploit the expected temporal correlation in streaming data.

This attack can be efficiently solved using convex optimization libraries such as CVXPY [17]; which ensures that the solution converges to the global optimum. The reconstructed signals $\hat{\mathbf{y}}_i(t)$ are then used to compute various metrics that quantify utility and privacy. This provides a rigorous, adversary-aware evaluation of privacy leakage under APBA and baseline mechanisms.

4.4.4 Evaluation Metrics

We use the following metrics for the quantification of utility and privacy.

- Mean Squared Error (MSE) and Mean Absolute Error (MAE) for utility.
- Breach count, average displacement, and resemblance as described in Section 3.5.3 in a CVXPY reconstruction attack. Breach count has been slightly relaxed to be the average number of times the sanitized tuple can be reconstructed to lie within a $\pm 10\%$ neighbourhood of the raw tuple.

The trade-off between cumulative privacy loss and estimation error has been shown using Pareto curves.

4.5 Results

4.5.1 APBA Epsilon Allocation Across timesteps

We first analyze how the proposed APBA mechanism allocates privacy budget across 1000 timesteps. Figure 4.2 shows the resulting allocation profiles for two real-world datasets. The total privacy budget ($\epsilon_i^{\text{total}}$) is taken to be 1 for both datasets.

By design, APBA increases the instantaneous per step $\epsilon_i(t)$ whenever the relative weight of the agent increases faster than the depletion of its remaining budget. This condition binds $\epsilon_i(t)$ to temporal variations in local uncertainty $u_i(t)$ and influence $\Delta_i(t)$ on the parameter being estimated.

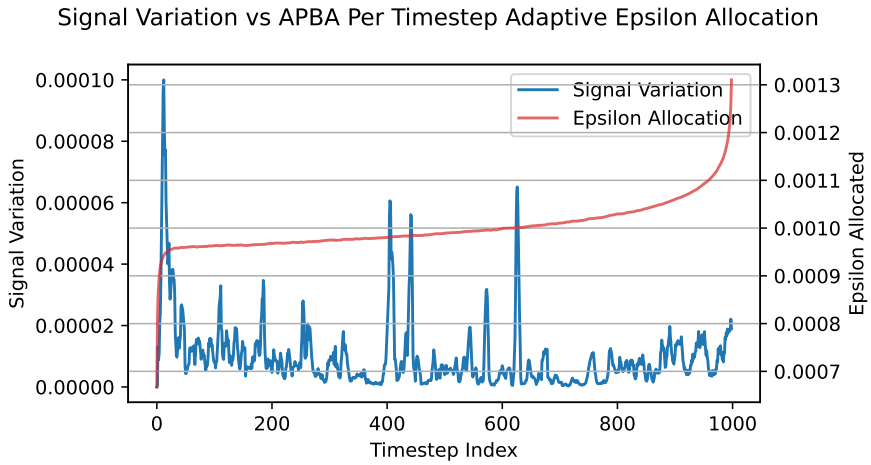
The Intel dataset exhibits a strongly non-stationary pattern with transient spikes in measurement variance. As a result, $u_i(t)$ and $\Delta_i(t)$ grow in the later timesteps, driving the relative weight of the agent at the t-th timestep $w_i(t)/W(t)$ upwards despite budget depletion. Accordingly, $\epsilon_i(t)$ shows a gradual rise followed by a sharp end-of-horizon spike, as APBA concentrates the remaining budget on the most informative timesteps. This adaptivity reduces noise during rapid state changes, improving the fidelity of the aggregate estimate.

In contrast, the NREL dataset is nearly stationary, with constant variance and influence over time. Here, $w_i(t)/W(t) \approx 1/S$ for all t, reducing the APBA to approximately uniform budget allocation:

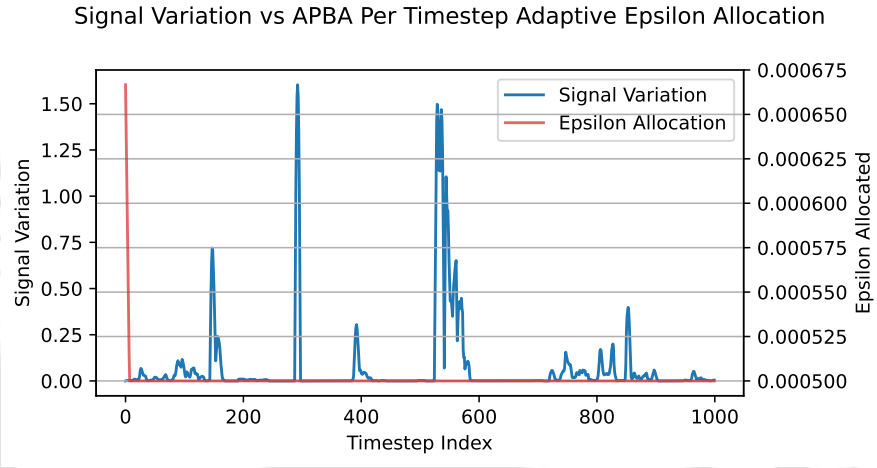
$$\epsilon_i(t) \approx \frac{\epsilon_i^{\text{total}}}{t^*}, \quad \forall t. \tag{4.2}$$

The resulting flat profile in Figure 4.2b confirms that APBA does not over allocate budget in stable periods, conserving resources while maintaining consistent privacy protection.

These findings validate APBA's design; it automatically spends more budget in high impact, high uncertainty regimes, and reverts to uniform spending when



(a) Intel Sensor Dataset.



(b) NREL Wind Turbine Dataset.

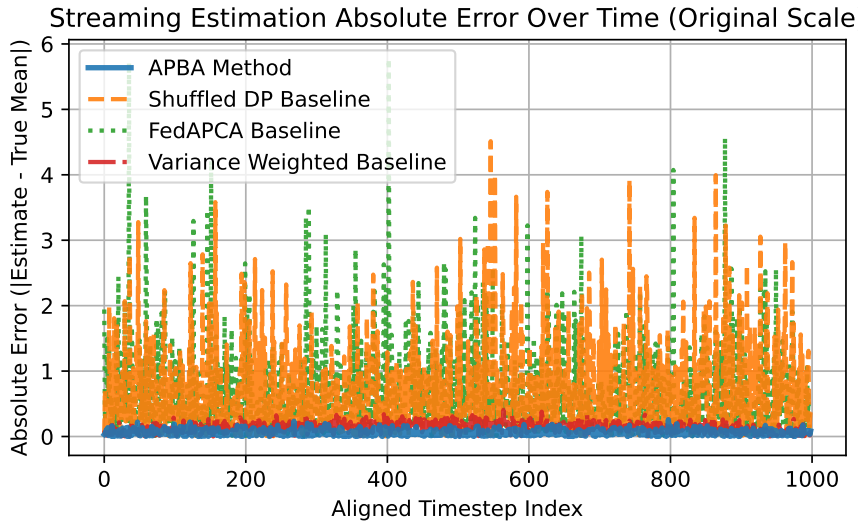
Figure 4.2: Epsilon allocation per timestep.

the signal is stationary. This adaptivity is crucial for balancing accuracy with differential privacy guarantees under sequential composition.

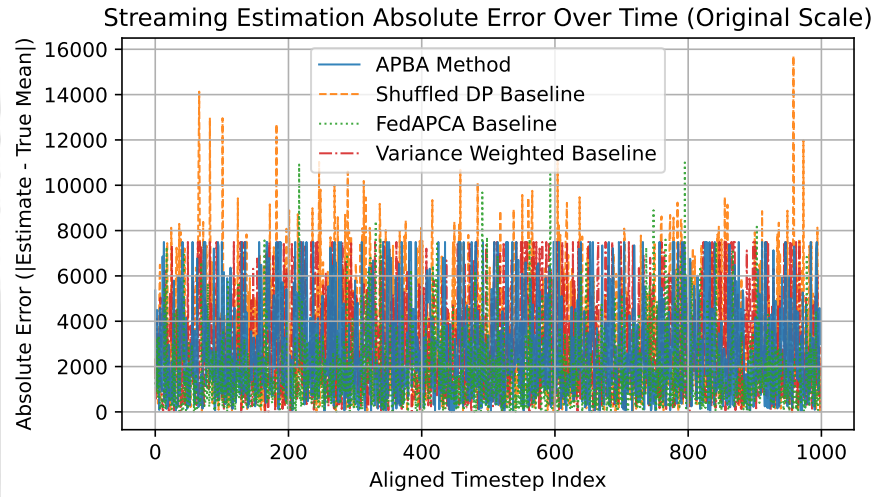
4.5.2 Estimation Accuracy

Next, we examine how adaptive budget allocation translates into estimation performance. Figure 4.3 compares the absolute error of APBA with two baselines: (i) Shuffled DP, which applies fixed per step ϵ followed by a shuffler, (ii) FedAPCA, a centralized mechanism with fixed ϵ applied to aggregated reports, and (iii) variance weighted baseline, where the privacy budget is allocated according to the empirical variance of the data.

For the Intel dataset (Figure 4.3a), APBA achieves the lowest error in all timesteps, reducing cumulative MAE by more than 90% compared to Shuffled DP and by roughly the same margin compared to FedAPCA. This gain comes from the APBA strategy of allocating more budget during periods of high variance



(a) Intel Sensor Dataset.



(b) NREL Wind Turbine Dataset.

Figure 4.3: Absolute error per timestep.

(Figure 4.2a), thus injecting less noise when observations are most informative. The Shuffled DP baseline suffers from constant noise injection, leading to frequent large deviations from the true mean. FedAPCA performs better than Shuffled DP but remains less accurate, as its centralized noise addition does not exploit local signal heterogeneity.

For the NREL dataset (Figure 4.3b), the performance gap is smaller but still positive. APBA reduces MAE by approximately 12% relative to shuffled DP. Since the dataset is stationary, APBA's privacy allocation becomes nearly uniform (Figure 4.2b). Thus, the benefit of adaptivity is less pronounced but remains non-negative. These results confirm that APBA delivers consistent utility guarantees in varying signal dynamics.

The performance of our proposed method is comparable to the variance

weighted baseline. The VW baseline allocates the per timestep budget independently, without explicitly enforcing a cumulative privacy constraint. Consequently, it does not guarantee that an agent's total privacy loss remains bounded by a predefined budget over time. In streaming MASs, where data is continuously released and temporal correlations amplify inference risks, failing to control cumulative privacy loss leads to unintended privacy leakage. In particular, repeated releases with variance-based allocation can exhaust privacy budget unpredictably or exceed acceptable long-term privacy guarantees. In contrast, APBA operates under a strict total privacy budget for each agent and dynamically distributes this budget across time while ensuring that the cumulative privacy loss never exceeds the limit. This makes APBA consistent with DP composition principles and suitable for streaming settings. By jointly considering instantaneous signal uncertainty and remaining privacy budget, APBA balances short-term utility with long-term privacy preservation. Therefore, while both methods achieve similar utility, APBA's formal cumulative privacy guarantees make it more principled and better aligned with the requirements of streaming MASs.

4.5.3 Result Summary

Tables 4.1 and 4.2 summarize performance across all metrics with each result reported as the mean over 30 independent Monte Carlo runs along with its corresponding 95% confidence interval. In the Intel dataset, APBA reduces MSE by over 99% relative to S-DP and F-APCA baselines and its performance is comparable to VW baseline, confirming that adaptive allocation can achieve near perfect tracking under modest noise levels. The higher breach count of APBA and VW reflects an expected privacy-utility trade-off. More accurate reconstructions are inherently closer to the true trajectory. Importantly, APBA remains within the formal privacy budget, so this does not violate the DP guarantees. This guarantee improves the applicability of our method to streaming MASs, where both cumulative privacy and utility are required over multiple timesteps.

Metric	S-DP [42]	F-APCA [72]	VW [11]	**APBA**
MSE	2.10426 ± 0.0491	2.18741 ± 0.0449	0.01141 ± 0.0002	**0.01206 ± 0.0002**
MAE	1.02807 ± 0.0107	1.04010 ± 0.0095	0.08612 ± 0.0007	**0.08903 ± 0.0007**
Breach Count	0.156 ± 0.0054	0.111 ± 0.0066	0.552 ± 0.0051	**0.529 ± 0.0050**
Displacement	9.13322 ± 0.2084	9.83289 ± 0.1932	2.04582 ± 0.0175	**2.14398 ± 0.0177**
Resemblance	0.50875 ± 0.0038	0.48366 ± 0.0040	0.51345 ± 0.0050	**0.51498 ± 0.0033**

Table 4.1: Results on Intel Sensor Dataset

Metric	S-DP [42]	F-APCA [72]	VW [11]	**APBA**
MSE	7.47687 ± 0.1144	2.14059 ± 0.0442	6.3606 ± 0.0639	**5.8088 ± 0.0639**
MAE	2.12446 ± 0.0173	1.02235 ± 0.0093	1.9487 ± 0.0140	**1.8645 ± 0.0140**
Breach Count	0.066 ± 0.0035	0.107 ± 0.0055	0.074 ± 0.0048	**0.038 ± 0.0068**
Displacement	22.08927 ± 0.5592	13.20502 ± 0.2160	17.91282 ± 0.3638	**20.43107 ± 0.3512**
Resemblance	0.49919 ± 0.0053	0.51891 ± 0.0042	0.50712 ± 0.0038	**0.50081 ± 0.0032**

Table 4.2: Results on NREL Wind Turbine Dataset

The substantial reduction in MSE and MAE for the Intel dataset in case of APBA and VW is due to the volatile nature of the Intel dataset. Unlike APBA, fixed-(ϵ) mechanisms inject the same amount of noise at every step, which obscures signal during transient spikes. However, this gain is not universal. On the more stationary NREL dataset, improvements are modest ($\approx$ 12% MAE reduction relative to shuffled DP), confirming that the large gains reflect dataset specific conditions rather than unrealistic performance.

In the NREL dataset, APBA delivers competitive utility: MAE is reduced by ∼12% and ∼9% relative to shuffled DP and variance weighted baselines, respectively, and the breach count is reduced by ∼42% and ∼50% respectively. Although MSE is higher than FedAPCA (5.80 vs. 2.14), APBA maintains better privacy–utility balance by avoiding excessive budget consumption in stable periods and preserving temporal structure, as indicated by resemblance scores.

The overhead of APBA is minimal, requiring only lightweight per-step updates to uncertainty and influence weights.

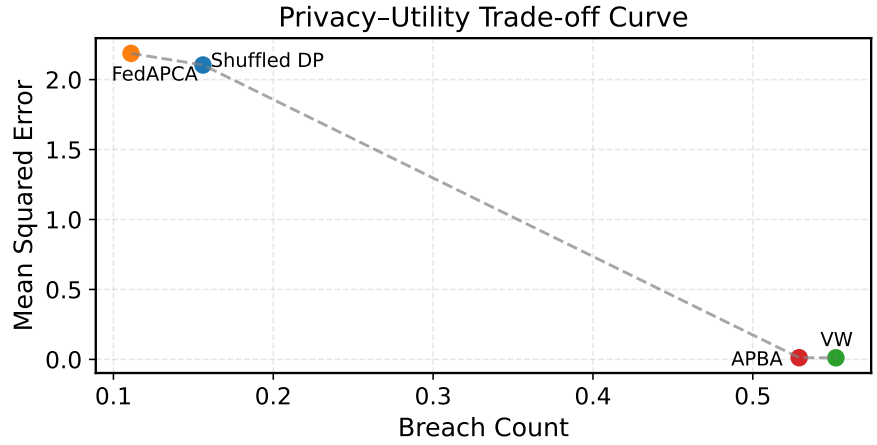
4.5.4 Pareto Analysis

Finally, we examine the Pareto frontier between privacy and utility (Figure 4.4). The Pareto curves present an empirical validation of the proposed allocation strategy. For the volatile Intel dataset, APBA lies near the bottom-right of the frontier, indicating high utility (low MSE) but with higher cumulative privacy loss, an intentional choice to minimize noise during transient spikes. For the smoother NREL dataset, APBA shifts toward the left-middle of the curve, achieving a lower privacy loss with only a modest utility improvement. This dataset specific behavior demonstrates that APBA dynamically adjusts its position along the privacy–utility frontier, prioritizing accuracy when needed and conserving privacy budget otherwise.

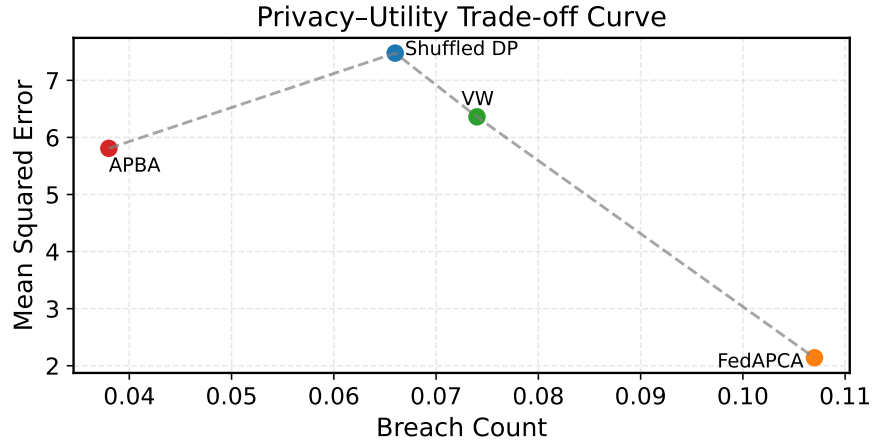
Overall, these findings demonstrate that adaptive privacy budget allocation can improve the privacy–utility trade-off in streaming multi-agent systems, while maintaining a bounded cumulative privacy budget. The result motivates further investigation for deployment in practical sensing and control applications.

4.6 Conclusion

We presented Adaptive Privacy Budget Allocation (APBA), a mechanism for dynamically distributing differential privacy budgets across timesteps in streaming MASs. Unlike static or client-level allocation schemes, APBA dynamically adjusts the per-step budget $\epsilon_i(t)$ based on local signal uncertainty and influence on the global estimate. Our theoretical analysis establishes that APBA guarantees the boundedness of the cumulative privacy loss of each agent. We assume a synchronized time model in which agents update privacy budgets concurrently at each round. At each timestep, every agent computes only its local uncertainty measure using local observations, which requires constant-time updates for running statistics and therefore incurs $\mathcal{O}(1)$ computational cost per agent. The influence scores are computed at the fusion center after collecting the agent reports. In the



(a) Intel Sensor Dataset



(b) NREL Wind Turbine Dataset

Figure 4.4: Privacy–utility Pareto frontier for Intel and NREL datasets.

straightforward implementation, evaluating the leave-one-out denominator computation for all S agents requires $\mathcal{O}(S)$ additional computations per timestep when the aggregation function is linear (e.g., mean estimation) at the fusion center. The computation of budget per timestep and the generation of noise is $\mathcal{O}(n)$. Since no pairwise coordination or cross-agent optimization is required, APBA is suitable for large-scale multi-agent sensing systems.

Preliminary theoretical analysis shows that APBA's performance is stable across variations in uncertainty weighting and budget scaling, which implies robustness without extensive hyperparameter tuning. Empirical results demonstrate that APBA substantially improves estimation accuracy in nonstationary environments. In the Intel dataset with high data variability, APBA concentrates privacy budgets on informative timesteps, reducing cumulative MAE by more than 90% relative to state-of-the-art baselines. In contrast, for the more stationary NREL dataset, APBA allocates budget almost uniformly over time, demonstrating its ability to handle data with low variability.

Part II

Dimension Reduction Methods

Chapter 5

Dynamic Projection Method: A Learning Based Approach

This chapter introduces a novel projection method, where the sanitization matrix is updated using the previous observation tuple and its sanitized counterpart. Due to the in place updation of the projection matrix using the data tuples, matrix regeneration costs are reduced and the method becomes data-dependent. [1]

[1]The contents of this chapter appear mostly in the paper titled "Dynamic Projection Method: A Learning Based Approach For Preserving Inference Privacy", in the Proceedings of IEEE GLOBECOM 2025 (CORE B Conference) and partly in "Dimension Reduction via Random Projection for Privacy in Multi-Agent Systems", which is currently under review.

In the previous two chapters, we studied the implementation of perturbation based methods including differential privacy (DP) based noise addition methods that can be used for the preservation of privacy in a MAS. However, we saw that the design and allocation of the DP budget needs to be handled carefully. Another class of mechanisms that can be used to achieve inference privacy is based on data compression. In this thesis, by compression, we imply a reduction in the dimension of the data tuple before it is shared with the fusion center. Thereafter, the fusion center employs a suitable inference model to combine these compressed representatives and deduce the system parameters.

Dimension reduction reduces computational complexity by representing the original data in a lower dimensional space. Operations such as storage, transmission, and learning algorithms become faster when they operate on fewer variables. By projecting the data onto a smaller set of informative components, the essential structure of the data can be preserved while eliminating irrelevant variations. This also leads to a degree of privacy protection. By transmitting only compressed representations instead of the full data vector, it becomes more difficult for an adversary to reconstruct the original data, thereby reducing the risk of inference attacks.

A prime example of a classical dimension reduction method is principal component analysis (PCA) [26]. In PCA, the principal components of the data are identified, and each data tuple is projected onto a subspace spanned by the principal components. However, identification of the principal components involve the computation of eigen values and eigen vectors which require the availability of the whole or a substantial part of the dataset. Once the components are identified, subsequent data tuples can be sanitized without the requirement of the dataset as a whole. One of the main privacy mechanism design requirements in a MAS is that the sanitization process should be computationally easy and should not require the whole or part of the dataset before application. Thus, application of PCA to streaming MAS settings is infeasible.

Projection based techniques are widely used not only for dimensionality reduction but also for improving computational efficiency in learning and inference problems such as robust concept deduction. In the robust concept problem, positive examples are combined to form the concept with highest possible accuracy. To reduce the overheads of computation and communication, random projection methods project the higher dimensional examples to a lower dimensional space by matrix multiplication. In basic random projection (BRP), the matrix used for projection is orthonormal and is generated from an appropriately chosen probability distribution. Initially introduced for reducing the dimension of each example in the robust concept deduction problem, BRP [67] uses a fixed orthonormal matrix

for all projections. Exploiting the analogy between the robust concept deduction problem and the MAS privacy problem, we could use BRP for the enforcement of privacy. However, the use of a single projection matrix for all the data tuples of an agent would lead to degraded privacy over time. Neuronal random projection (NRP) [34] is a novel modification of BRP which uses a new matrix with independent entries from a bounded distribution for each sanitization step. Due to the enhanced randomness introduced due to the use of a new matrix at each sanitization step, the resultant privacy increases. However, these methods suffer from a few limitations. The cost of orthonormalization in BRP makes it infeasible for use in a MAS. On the other hand, the NRP method does not take into account the inherent properties of the underlying data. The matrix generation method is completely data independent unless an additional constraint on utility or privacy is enforced. To counter these limitations, we propose a learning based projection method, named dynamic projection method (DPM), which also uses a new matrix at each step, but does not regenerate it from a distribution. It instead updates the existing matrix using the current observation and its sanitized counterpart, thereby obtaining a fresh matrix for the next step. This not only saves resources but also decreases the data processing time before communication. We further incorporate a learning rate and a regularization factor for the matrix updation process which affect the tradeoff and can be tuned according to requirement. This method balances both utility and privacy without the need of any additional constraint, making it a suitable choice for MAS. We introduce our proposed algorithm in this chapter and illustrate the efficiency of our mechanism by thorough experimentation.

The organization of the chapter is as follows. The problem definition is outlined in Section 5.1. The problem of robust concept deduction and the BRP and NRP mechanisms are discussed in Section 5.2. The proposed DPM methodology is described in Section 5.3 along with the theoretical guarantees of utility and privacy. The thorough experimentation is illustrated in Section 5.4.

5.1 Problem Definition

We consider the MAS model introduced in Section 1.2, in which S autonomous agents send their observations to a central fusion center about a system parameter tuple at regular intervals. The structure of the observation tuple of the i-th agent at the t-th step as mentioned in Equation 4.1 is

$$\mathbf{y}_i = (p_{1t}, p_{2t}, \ldots, p_{kt}, g_{1t}, \ldots, g_{(n-k)t}) \in \mathbb{R}^n. \tag{5.1}$$

Upon implementation of a decentralized sanitization function, the data tuple becomes

$$\mathcal{M}(\mathbf{y}_i) = (p'_{1t}, p'_{2t}, \ldots, p'_{k't}, g'_{1t}, \ldots, g'_{(m-k')t}) \in \mathbb{R}^m, \tag{5.2}$$

where $m \leq n$.

In this paper, we propose a decentralized, matrix multiplication based dimension reduction method for sanitization of the data by each agent. This method uses the previous data tuple and its sanitized counterpart to update the transformation matrix.

5.2 Preliminaries

In this section, we introduce the robust concept problem and subsequently, the BRP and NRP mechanisms. The initialization of the sanitization matrix in our proposed mechanism can be done by the use of either of the two mechanisms.

5.2.1 Robust concepts

We formally define the notion of a concept, followed by the definition of a robust concept, and draw parallels with the MAS problem.

Definition 5.1 *(**Concept** [67]) A concept is an idea of a physical or abstract phenomenon. It is formed by combining examples that contribute to understanding the attributes of the concept. The examples are combined following some preset rules.*

For example, if $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_S}$ are positive examples for a concept $\mathbf{x}$, then

$$\mathbf{x} = F(\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_S}),$$

where F is the function used to deduce the concept. In a MAS setup, the system parameter which the fusion center wishes to deduce is analogous to a *concept*. In order to deduce the concept, the fusion center gathers and uses the *examples*, which in this case are the agent observations. For each such example, the data tuple elements are the *attributes* of the example.

Definition 5.2 *(**Robust Concept** [67]) Let $\mathbf{y}_i$ be the actual data and $\tilde{\mathbf{y}}$ be the modified data. Let ∇_p (> 0) be a robustness parameter. Consider $F(\mathbf{y}_i)$ and $F(\tilde{\mathbf{y}})$ to be the concepts deduced from $\mathbf{y}_i$ and $\tilde{\mathbf{y}}$, respectively. The concept is said to be ∇_p-robust if for $\nabla_p > 0, \exists\, \nu > 0$ such that*

$$||F(\mathbf{y}_i) - F(\tilde{\mathbf{y}})|| < \nu \;\; \textit{whenever} \;\; 0 < ||\mathbf{y}_i - \tilde{\mathbf{y}}|| < \nabla_p.$$

A concept is said to be ∇_p robust if it can be accurately inferred even when the input example is perturbed within a bounded range defined by ∇_p. That is, small deviations, such as minor reductions in the number of attributes or slight modifications to attribute values, do not substantially affect the correctness of the inferred concept. In the context of a MAS, the agent observations may be subject to local modifications (e.g., due to decentralized privacy-preserving transformations), yet the fusion center must still be able to reliably deduce the parameter.

Robustness is a property inherent to the concept from an estimation perspective. The robustness parameter governs the tolerance to input perturbations and influences the success or failure of concept inference.

5.2.2 Basic random projection

BRP [67] is a dimensionality reduction mechanism where an n-tuple data vector is multiplied with an $n \times m$ uniform random orthonormal matrix A to get an m-tuple. A fixed matrix is used for the projection of all data tuples. Using this mechanism, $\mathcal{M}(\mathbf{y}_i)$ can be depicted mathematically as

$$\mathcal{M}(\mathbf{y}_i) = A^T \mathbf{y}_i. \tag{5.3}$$

One important property of BRP is that it preserves pairwise distances when the matrix A is a uniform random orthonormal matrix. Using the lemma of Johnson and Lindenstrauss [30], Vempala[67] has shown that the distance between two pairs of projections is bounded, with high probability, by a factor of the distance between the original data points. We have reformulated the same as Lemma 5.1 to highlight the similarity between distance preservation and differential privacy guarantee.

Lemma 5.1 *Let $\mathcal{S} \subset \mathbb{R}^n$ be a set of data points with $|\mathcal{S}| = S$, where each point corresponds to the data from an agent. For any $\gamma \in (0, 0.405)$, consider an uniform random projection to an m-dimensional subspace, where m is suitably chosen, the following holds:*
For every pair $\mathbf{y}_i, \mathbf{y}_j \in \mathcal{S}$,

$$\begin{aligned} Pr(e^{-\gamma}||\mathbf{y}_i - \mathbf{y}_j||^2 \leq ||\mathcal{M}(\mathbf{y}_i) - \mathcal{M}(\mathbf{y}_j)||^2 \leq e^{\gamma}||\mathbf{y}_i - \mathbf{y}_j||^2) \\ \geq 1 - 2\sqrt{m}e^{-(m-1)(\frac{\sinh^2\gamma}{4} - \frac{\sinh^3\gamma}{6})}. \end{aligned} \tag{5.4}$$

where $\mathcal{M}(\mathbf{y}_i)$ and $\mathcal{M}(\mathbf{y}_j)$ are the projections of $\mathbf{y}_i$ and $\mathbf{y}_j$ respectively.

Following a result due to Vempala [67], if we choose the projection dimension

m such that

$$m \geq \frac{9 \ln S}{\sinh^2 \gamma - \frac{2}{3} \sinh^3 \gamma} + 1,$$

and substitute in the right-hand side of inequality 5.4, we have

$$1 - 2\sqrt{m} e^{-(m-1)(\frac{\sinh^2 \gamma}{4} - \frac{\sinh^3 \gamma}{6})} \geq \frac{1}{2}. \tag{5.5}$$

This result implies that the pairwise distances between projected data points are preserved within a multiplicative factor of $e^{\pm\gamma}$ with probability of at least $\frac{1}{2}$.

The distance preserving property of random projection ensures that the data vectors close to each other do not get mapped to vectors far apart in the lower dimensional space. We assume that the metric used for the calculation of pairwise distances is the Euclidean metric. This property is critical to maintaining the utility of sanitized data. However, the generation of an orthonormal matrix required for such projections incurs significant computational cost, typically on the order of $\mathcal{O}(n \times m^2)$ [28]. Given the limited computational capacity and energy constraints of agents in a MAS, it may not always be feasible for each agent to independently generate and apply a random orthonormal projection for data sanitization. To address this limitation, we next introduce a modification of the BRP technique that retains a similar distance-preserving property, but is considerably more efficient and practical for deployment in resource-constrained environments.

5.2.3 Neuron-friendly or neuronal random projection

It has been established that, for the purpose of dimensionality reduction, it is sufficient to use random matrices whose entries are independently drawn from a bounded distribution [67]. Such matrices, though not orthonormal, are capable of approximately preserving pairwise distances. Lemma 5.2 formalizes the distance preserving property. More specifically, like the previous lemma, it shows that for any pair of input vectors, their squared Euclidean distance is preserved up to a multiplicative factor $e^{\pm\gamma}$.

Lemma 5.2 *Let $\mathbf{y}_i, \mathbf{y}_j \in \mathbb{R}^n$. Let $\mathcal{M}(\mathbf{y}_i)$, $\mathcal{M}(\mathbf{y}_j)$ be the projections of $\mathbf{y}_i$ and $\mathbf{y}_j$ to $\mathbb{R}^m$ via a random matrix A with independent entries from a bounded distribution. Then*

$$\begin{aligned} Pr(e^{-\gamma}||\mathbf{y}_i - \mathbf{y}_j||^2 \leq ||\mathcal{M}(\mathbf{y}_i) - \mathcal{M}(\mathbf{y}_j)||^2 \leq e^{\gamma}||\mathbf{y}_i - \mathbf{y}_j||^2) \\ \geq 1 - 2e^{-(\sinh^2 \gamma - \sinh^3 \gamma)\frac{m}{4}}. \end{aligned} \tag{5.6}$$

For $\gamma \in (0, 0.405)$, the probability of pairwise distance preservation is higher in Lemma 5.2 as compared to the preceding lemma.

To address the computational limitations associated with generating orthonormal matrices in standard random projection, an alternative approach is NRP [34]. The nomenclature NRP draws inspiration from human anatomy, wherein neurons (analogous to agents) communicate information to the brain (analogous to the fusion center) to deduce and disseminate complex concepts. In this analogy, the generation of an orthonormal matrix by each neuron would represent a computationally intensive task, much like the high resource demands imposed on agents. In contrast, the NRP method is computationally lightweight, thereby offering a neuron-friendly alternative that is both privacy-preserving and efficient.

In NRP, each agent locally generates a random projection matrix with independent entries drawn from a bounded distribution. Unlike the BRP method, where an agent may use the same compression matrix across multiple data tuples, the NRP approach involves generating a fresh projection matrix for each sanitization instance. This per-instance randomness enhances the unpredictability of the transformation, thereby contributing to a stronger privacy guarantee.

Based on Lemma 5.1 and Lemma 5.2, in Theorem 5.1, we also establish that the NRP method provides the same privacy guarantee as that of conventional random projection.

Theorem 5.1 *The probabilistic guarantee for neuronal random projection to satisfy the distance preservation property is as good as standard random projection.*

Proof. Let the probability of satisfying the distance preservation property be equal when m is m_1 for basic random projection and m_2 for neuronal random projection.
Solving the following equation:

$$1 - 2\sqrt{m_1}e^{-(m_1-1)(\frac{\sinh^2\gamma}{4}-\frac{\sinh^3\gamma}{6})} = 1 - 2e^{-(\sinh^2\gamma-\sinh^3\gamma)\frac{m_2}{4}},$$

we have

$$m_2 = \frac{(m_1-1)(\sinh^2\gamma - \frac{2}{3}\sinh^3\gamma) - 2\ln m_1}{\sinh^2\gamma - \sinh^3\gamma}.$$

Since $\gamma \in (0, 0.4054651081)$, we have $m_2 < 1.243m_1$. Since m_2 is the reduced dimension, it has to be an integral value. This implies, $m_2 \leq m_1$. □

In this chapter, we propose a similar decentralized, matrix multiplication based dimension reduction method for sanitization of the data by each agent. However, unlike NRP, our proposed method does not regenerate the sanitization matrix before each sanitization step. Rather, this method learns from the existing data tuple and its sanitized representative to update the current matrix for the next sanitization step. The method thus obtained, becomes a data independent dimension reduction mechanism.

5.3 Dynamic projection method

Our proposed method is a learning based projection method. The projection is done by matrix multiplication. However, unlike standard projection based methods, our method neither regenerates nor uses the same matrix for each data transmission. We update the projection matrix using the raw and sanitized data obtained after each sanitization step. Since the matrix gets updated after each data sanitization step, our method is dynamic in its approach, and thereby called DPM. A learning rate and a regularization factor affect the amount of modification of the projection matrix. They directly influence the effect of each raw and sanitized tuple on the new matrix. These factors also affect the privacy-utility tradeoff. We now outline our method, for one agent. The steps involved are illustrated in Figure 5.1.

Step 1 *The agent generates a matrix* $A_0 \in \mathbb{R}^{n\times m}$ *as in BRP [67] or NRP [34].*

Step 2 *Let the observations of the i-th agent be a sequence of vectors* $\{\mathbf{y}_t\}_{t=1}^{t^*}$ *with* $\|\mathbf{y}_t\|_2 \leq R$. *For each time step* $t \geq 1$, *the sanitization and matrix updation steps are as follows:*

$$\begin{cases} \tilde{\mathbf{y}}_t = A_{t-1}^T \mathbf{y}_t, \\ A_t = A_{t-1} + \eta(\mathbf{y}_t \tilde{\mathbf{y}}^T - \lambda A_{t-1}), \end{cases} \tag{5.7}$$

where, η *is the learning rate and* λ *is the regularization term. The matrix is reorthogonalized every* T_r *steps, and* A_t *is reinitialized from the same distribution every* T_f *steps.*

Conditioned on the realization of A_{t-1}, the released output $\tilde{\mathbf{y}}_t = A_{t-1}^T \mathbf{y}_t$ is deterministic. Marginally, however, $\tilde{\mathbf{y}}_t$ is a random vector whose distribution is induced by the random initialization of A_0 together with the adaptive evolution of the projection matrix.

Each agent uses the same method for data sanitization in a decentralized manner. A lower value of $\eta \in [0.01, 0.1]$ will result in stability of the process and will ensure that the matrix does not undergo a drastic change during the updation process. It will also ensure that the influence of new data points on the matrix is not large. A value of $\eta = 0.1$ ensures faster learning as well as stability. Similarly, λ acts as a decay factor and prevents overfitting of the matrix to recent observation data. A moderate value of $\lambda \in [0.1, 1]$ would ensure that new datapoints influence the new matrix while retaining past structure. We further illustrate our method using a numerical example.

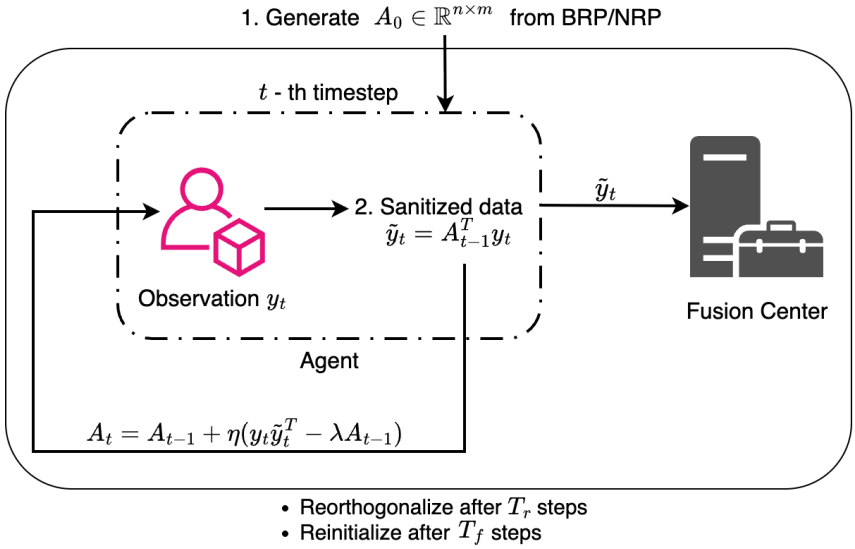


Figure 5.1: Proposed Dynamic Projection Methodology.

Example 5.2 *For a particular agent, we take* $\{\mathbf{y}_t\}_{t=1}^{t^*} \in \mathbb{R}^5$ *and* $m = 3$. *We also assume* $\eta = 0.1$ *and* $\lambda = 1$. *The initial matrix* A_0 *is randomly generated as in NRP. Let*

$$A_0 = \begin{bmatrix} 0.1 & -0.2 & 0.3 \\ 0.4 & 0.0 & -0.5 \\ -0.1 & 0.6 & 0.2 \\ 0.3 & -0.4 & 0.0 \\ -0.2 & 0.1 & -0.3 \end{bmatrix}.$$

Let the observation vector $\mathbf{y_1}$ *be*

$$\mathbf{y_1} = [2, -1, 0.5, 1, -0.5]^T.$$

Then the first sanitized tuple $\tilde{\mathbf{y}}_1$ *is*

$$\tilde{\mathbf{y}}_1 = A_0^T \mathbf{y_1} = [0.15, -0.55, 1.35]^T.$$

Therefore, $A_1 = A_0 + \eta(\mathbf{y_1}\tilde{\mathbf{y}}_1^T - \lambda A_0)$ *is*

$$A_1 = \begin{bmatrix} 0.120 & -0.290 & 0.540 \\ 0.345 & 0.055 & -0.585 \\ -0.0825 & 0.5125 & 0.2475 \\ 0.285 & -0.415 & 0.135 \\ -0.1875 & 0.1175 & -0.3375 \end{bmatrix}.$$

As we can see from the example, the in-place updation of the matrix introduces randomness. Due to the incorporation of the learning rate and regularization factor, the entries in the matrix stay within a range. The matrix is reorthogonalized at regular intervals and refreshed before the variance of the matrix entries cross a prespecified limit.

The following Theorem 5.3 shows that by updating our initial projection matrix with a small learning rate, regularization, periodic reorthogonalization, and occasional full refresh, we achieve a tradeoff between utility and privacy. The randomness in the proposed mechanism arises solely from the initialization of the projection matrix A_0, whose entries are independently sampled from the bounded distribution described in Section 5.2. Conditioned on A_0, the subsequent matrix updates are deterministic functions of the observed data. Consequently, all probabilities in the following analysis are taken with respect to the randomness of the initial projection matrix and the induced sequence of projection matrices.

- The pairwise inner products are almost as accurate as possible, distorting distances by only $\mathcal{O}\left(\frac{R^2}{\sqrt{m}} + \eta R^2 t\right)$. This guarantees that by choosing m appropriately and keeping ηt small, the geometry of the data under evolving projection is preserved. This eventually means that the data utility gets retained.
- The total differential privacy loss over T_f is bounded, so privacy leakage stays within a preset budget.

Theorem 5.3 *For an agent, let $\eta \leq \frac{1}{\lambda + \|\mathbf{y}_t\|_2^2}$ and $\eta\lambda < 1$. The sanitization process outlined in Equation 5.7 satisfies*

(a) Utility Guarantee: *For all $t \leq T_f$ and any fixed $\mathbf{y_1}, \mathbf{y_2} \in \mathbb{R}^n$ with $\|\mathbf{y_1}\|_2, \|\mathbf{y_2}\|_2 \leq R$, the inner product distortion is bounded with high probability:*

$$\left|\langle A_t^T \mathbf{y_1}, A_t^T \mathbf{y_2}\rangle - \langle \mathbf{y_1}, \mathbf{y_2}\rangle\right| \leq \mathcal{O}\left(\frac{R^2}{\sqrt{m}} + \eta R^2 t\right).$$

(b) Privacy Guarantee: *Considering neighbouring input sequences such that $\|\mathbf{y}_t - \mathbf{y}'_\mathbf{t}\|_2 \leq \Delta$ with $\Delta \leq 2R$, the cumulative privacy leakage from adaptivity over T_f steps satisfies*

$$\epsilon_{total} = \mathcal{O}\left(\frac{\eta R^2 T_f}{\sigma}\right), \quad \delta = \textit{negligible},$$

so the mechanism is $(\epsilon_{total}, \delta)$-differentially private. Here, σ is the variance of the entries in the projection matrix.

Proof. We prove (a) and (b) separately.

Part (a): Utility guarantee

From the Johnson-Lindenstrauss (JL) [14] lemma we have: for any fixed vectors $\mathbf{y_1}, \mathbf{y_2} \in \mathbb{R}^n$, there exists a constant $C > 0$ (depending on the distribution parameters) such that

$$Pr\Big[\Big|\langle A_0^T \mathbf{y_1}, A_0^T \mathbf{y_2}\rangle - \langle \mathbf{y_1}, \mathbf{y_2}\rangle\Big| \geq \epsilon\Big] \leq 2\exp\Big(-\frac{m\epsilon^2}{C\,R^4}\Big).$$

Thus, by choosing $\epsilon = \mathcal{O}(R^2/\sqrt{m})$, the distortion caused by using A_0 is small with high probability.

Unrolling the recursive update rule as in Equation 5.7, we have, for any fixed vector $\mathbf{y} \in \mathbb{R}^n$,

$$A_t^T \mathbf{y} = (1-\eta\lambda)^t\, A_0^T \mathbf{y} + \eta \sum_{i=1}^{t} (1-\eta\lambda)^{t-i}\, (\mathbf{y}_i\, \tilde{\mathbf{y}}_i^T)^T \mathbf{y}. \tag{5.8}$$

When comparing the inner product $\langle A_t^T \mathbf{y_1}, A_t^T \mathbf{y_2}\rangle$ to $\langle \mathbf{y_1}, \mathbf{y_2}\rangle$, two sources of error emerge:

1. The error from the initial projection, which is $\mathcal{O}(R^2/\sqrt{m})$.
2. The error from the adaptive updates. Each update introduces a perturbation of the form $\eta\, \mathbf{y}_i\, \tilde{\mathbf{y}}_i^T$. Since $\|\mathbf{y}_i\|_2 \leq R$ and (by design, due to reorthogonalization) $\|\tilde{\mathbf{y}}_i\|_2 = \|A_{i-1}^T \mathbf{y}_i\|_2$ is also $\mathcal{O}(R)$, we have

   $$\|\eta\, \mathbf{y}_i\, \tilde{\mathbf{y}}_i^T\|_2 \leq \eta R^2.$$

   Therefore, over t steps, the cumulative drift due to updates is at most $\eta R^2\, t$.

Hence, the total deviation in the preservation of inner products is bounded by

$$\Big|\langle A_t^T \mathbf{y_1},\ A_t^T \mathbf{y_2}\rangle - \langle \mathbf{y_1}, \mathbf{y_2}\rangle\Big| \leq \mathcal{O}\Big(\frac{R^2}{\sqrt{m}} + \eta R^2 t\Big). \tag{5.9}$$

The JL guarantee applies only to the randomly initialized matrix A_0. The adaptive updates introduce an additional perturbation whose cumulative magnitude is bounded by $\mathcal{O}(\eta R^2 t)$. Therefore the utility guarantee should be interpreted as a perturbation of the original JL embedding rather than a direct application of the classical JL lemma.

Part (b): Privacy guarantee

At each time step t, the mechanism outputs

$$\tilde{\mathbf{y}}_t = A_{t-1}^T \mathbf{y}_t.$$

Consider two neighbouring inputs $\mathbf{y}_t$ and $\mathbf{y}'_t$ such that $\|\mathbf{y}_t - \mathbf{y}'_t\|_2 \leq \Delta$.

The change in the output is

$$\|A_{t-1}^T(\mathbf{y}_t - \mathbf{y}'_t)\|_2 \leq \|A_{t-1}^T\|_2 \, \Delta.$$

Under the reorthogonalization assumption, the operator norm $\|A_{t-1}^T\|_2$ is maintained at approximately $\sqrt{m}$. Thus, the per-step sensitivity is bounded by

$$\Delta_{\text{step}} \leq \sqrt{m}\, \Delta.$$

Since the rows of A_0 are drawn as in random projection and similar properties persist (by reorthogonalization) for A_{t-1}, the output $\tilde{\mathbf{y}}_t$ is essentially obtained via a randomized linear mapping. Using the above sensitivity bound together with the randomized projection mechanism, the per-step privacy loss can be bounded in terms of the projection sensitivity and the variance parameter σ. Consequently, the effective privacy contribution at each step is proportional to

$$\epsilon_t = \mathcal{O}\left(\frac{\Delta_{\text{step}}}{\sigma}\right) = \mathcal{O}\left(\frac{\Delta\sqrt{m}}{\sigma}\right).$$

The adaptive update rule introduces an additional data-dependent perturbation at each step of size at most ηR^2. Thus, using basic and advanced composition techniques such as the moments accountant, the effective privacy loss per step can be bounded by an amount proportional to $\eta R^2/\sigma$. Over T_f steps, the total privacy loss is then

$$\epsilon_{\text{total}} = \mathcal{O}\Big(\frac{\eta R^2 T_f}{\sigma}\Big).$$

Since the mechanism is refreshed every T_f steps, the cumulative privacy leakage does not become unbounded over time, and the corresponding δ is kept negligible (for instance, $\delta \leq \exp(-\Omega(m))$).

□

Corollary 5.4 *Under the assumptions of Theorem 1 (DPM),*

1. *we reorthogonalize A_t every T_r (reorthogonalization frequency) steps to enforce $\|A_t^T A_t - I\|_2 \leq \tau$, and*

2. *we refresh A_t every T_f (refresh frequency) steps to reset the privacy leakage and maintain $\epsilon_{\text{total}} \leq \epsilon_{\max}$.*

Then it suffices to choose

$$T_r = \mathcal{O}\big(1/(\eta\lambda)\big), \text{ and } T_f = \mathcal{O}\Big(\frac{\epsilon_{\max}\,\sigma}{\eta\,R^2}\Big). \tag{5.10}$$

Proof. We prove the two parts separately.

1. We unroll the update rule and use

   $E_t = A_t^T A_t - I$. Since $A_0^T A_0 = I$, by induction it follows that

   $$E_t = (1-\eta\lambda)^{2t} E_0 \; + \; \frac{\eta R^2}{\lambda}\Big(1-(1-\eta\lambda)^{2t}\Big)\, H_t,$$

   where $\|H_t\|_2 \leq 1$.
   Hence

   $$\|E_t\|_2 \;\leq\; \frac{\eta R^2}{\lambda}\big(1-(1-\eta\lambda)^{2t}\big).$$

   To maintain $\|E_t\|_2 \leq \tau$, we require

   $$1-(1-\eta\lambda)^{2t} \;\leq\; \frac{\tau\,\lambda}{\eta R^2}.$$

   Since $\log(1-\eta\lambda) \approx -\eta\lambda$ for small $\eta\lambda$, this yields

   $$t \;\lesssim\; \frac{1}{2\eta\lambda}\,\log\Big(\frac{1}{1-\tau\lambda/(\eta R^2)}\Big) = \mathcal{O}\big(1/(\eta\lambda)\big).$$

   Thus, $T_r = \mathcal{O}(1/(\eta\lambda))$ suffices.

2. From the privacy analysis we have

   $$\epsilon_{\text{total}} = \mathcal{O}\Big(\frac{\eta R^2\, T_f}{\sigma}\Big).$$

   To enforce $\epsilon_{\text{total}} \leq \epsilon_{\max}$, we have $T_f \leq \frac{\epsilon_{\max}\,\sigma}{\eta R^2}$

   Thus, $T_f = \mathcal{O}(\epsilon_{\max}\,\sigma/(\eta R^2))$.

□

5.4 Experimentation and results

The objective of our algorithm is to ensure inference privacy, by ensuring resistance to reconstruction attacks, while preserving data utility. We observe that preventing the reconstruction of the original data significantly limits an adversary's ability

to infer sensitive secondary information. We demonstrate experimentally that, despite not incorporating explicit utility or privacy constraints, our method achieves an optimal balance between the two when compared to existing approaches. We assume an honest-but-curious adversary that observes the sanitized outputs transmitted by the agents and has full knowledge of the DPM algorithm, including the learning rate, regularization parameter, reorthogonalization schedule, and refresh policy. The projection matrix maintained by each agent is not transmitted and is therefore unavailable to the adversary. Consequently, reconstruction attempts are based solely on the released sanitized observations. We consider that the adversary uses a reconstruction based method to reconstruct the data. Since our method is a projection method, the attacker can use an appropriate inverse-projection method to reconstruct the original data tuple. For our experimentation, we have considered a matrix based reconstruction algorithm wherein the attacker generates an inverse matrix to obtain a reconstruction of the original datapoint. In our experiments, we compared the proposed approach with principal component analysis (PCA) [26] and the data independent NRP method [34].

5.4.1 Datasets used

In the experiments we used two datasets.

- Real life hospital dataset: We considered a patient record dataset from a hospital [9] collected over a two-year period. Each observation in the dataset is represented as a 50-dimensional tuple. The dataset was preprocessed by removing non-numeric fields and converting binary categorical fields (such as gender) into binary numeric values. All remaining numeric attributes were normalized. Among the 50 features, 14 are designated as private and the rest as public. Parameters that can be used for profiling, such as age, gender, and area of residence, are treated as private parameters, while health-related attributes like hemoglobin (HB), total leukocyte count (TLC), and platelet count are treated as public parameters. We assume that each agent (representing an individual patient or data source) sends multiple observations at different timestamps.

- Synthetic data: We consider a scenario where a single agent makes multiple observations over time. Each observation is represented as a 50-dimensional tuple, comprising 14 private parameters and 36 public parameters. The tuples are assumed to be generated at regular intervals, and each one is sanitized before being transmitted to a central fusion center. The elements of each tuple are independently and randomly drawn from a uniform distribution U(-0.5, 0.5), simulating uncorrelated and unbiased data generation.

5.4.1.1 Performance metrics

For quantifying and comparing the performance of the four mechanisms, we have used the following metrics, which are appropriately modified versions of the ones introduced in Section 3.5.3:

- Breach Count [34]: Average number of data points correctly reconstructed to be within a neighbourhood of the original data point. Mathematically, let I be an indicator variable that is set to 1 if the attacker correctly places the data tuple in a neighbourhood of the original tuple. Otherwise, the variable is set to 0. For the reconstructed dataset $\{a'_1, a'_2, \ldots, a'_{n'}\}$ and an actual dataset $\{a_1, a_2, \ldots, a_{n'}\}$ of size n', the breach count is defined as:

$$\text{breach count} = \frac{\sum_{i=1}^{n'} I_{(a'_i \in N_{a_i})}}{n'},$$

  where N_{a_i} is the neighbourhood of a_i. A point a'_i is said to belong to the neighbourhood N_{a_i} if the distance between the two points is at most equal to the neighbourhood radius. The neighbourhood radius is set according to the threshold prespecified. Higher value of breach count implies lower levels of privacy, as the agent is able to correctly reconstruct the datapoints.

  For our analysis, we define a breach as occurring when a reconstructed tuple falls within a $\pm 10\%$ range of the corresponding raw tuple, i.e., a threshold of 0.1 is used.

- Displacement [34]: This metric gives the average Euclidean distance between the original and the reconstructed data for all the agent observations taken together. It is measured in standard Euclidean distance units.

  For a given reconstructed dataset and the corresponding actual dataset, the displacement is defined as

$$\text{displacement} = \frac{\sum_{i=1}^{n'} \rho(a_i, a'_i)}{n'},$$

  where a'_i and a_i belong to the reconstructed and actual dataset, respectively. The displacement between two points is measured in terms of the Euclidean distance (i.e. the l^2 norm $||.||_2$). A lower displacement indicates higher utility, as it reflects lesser deviation from the original data.

- Resemblance [34]: Gives the average number of neighbours common to both the original and the reconstructed data point. More formally, let $N_R = \{\mathbf{s_1}, \ldots, \mathbf{s_k}\}$ be the set of k nearest neighbours corresponding to the actual data point. Similarly, let $N'_R = \{\mathbf{s'_1}, \ldots, \mathbf{s'_k}\}$ be the neighbours corresponding

to the reconstructed data point. Then resemblance is defined as

$$\frac{|N_R \cap N_R'|}{|N_R|}.$$

Lower resemblance implies higher levels of achieved privacy as the number of nearest neighbours that are common to both the actual and reconstructed data points is less in number. This would imply that an adversary is not able to correctly profile the agent using its neighbours.

For good privacy preservation and utility retention, we want all the above performance metrics to have low values simultaneously.

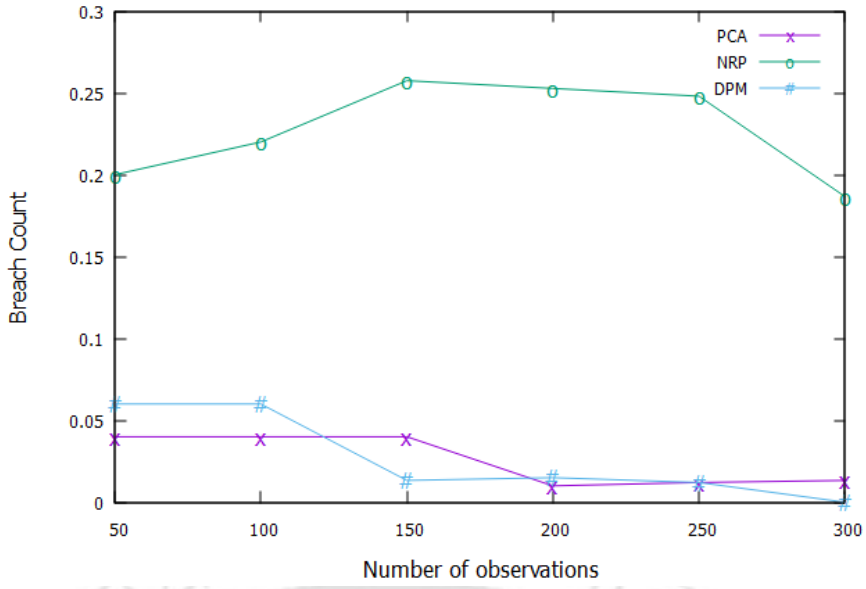
The objective of our algorithm is to ensure inference privacy by providing resistance against reconstruction attacks while preserving data utility. Accordingly, we evaluate privacy through reconstruction-attack metrics that quantify the adversary's reconstruction capability. These metrics provide empirical evidence of privacy preservation and are interpreted as measures of reconstruction resistance.

5.4.2 Results

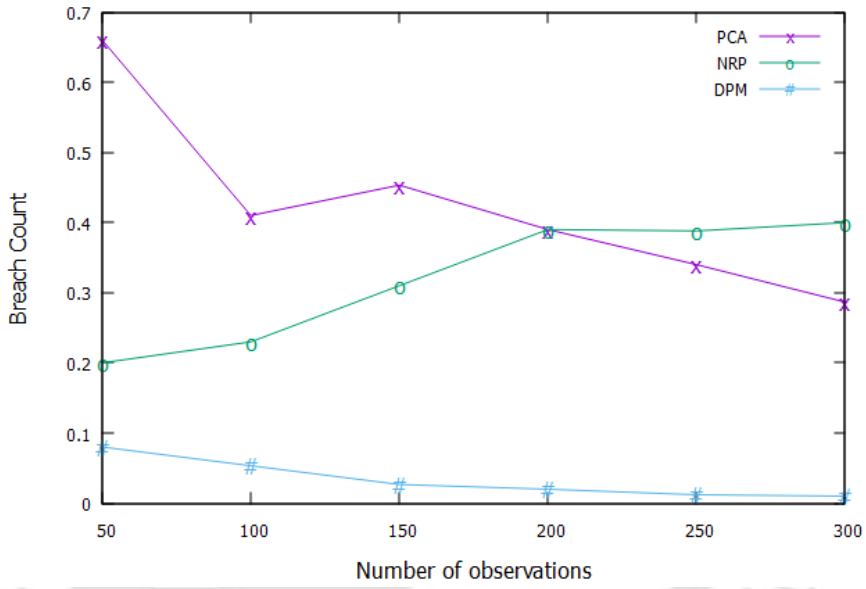
All experiments were repeated multiple times (about 100) and the average values are plotted. For NRP, we have taken $\epsilon = 0.6$. Other values of ϵ for NRP also yielded similar trends. We have used $\eta = 0.1$ and $\lambda = 1$ for DPM and considered an adversarial inverse projection reconstruction. All methods project to the same dimension, assume the same attacker knowledge, and use the same reconstruction model. We have repeated the reconstruction algorithm 100 times and taken the average metric values.

1. Breach Count analysis: As shown in Figure 5.2, DPM performs consistently well across both datasets. Its breach count decreases with more observations. This is due to DPM's evolving matrix, which incorporates previous data — unlike PCA, which is data-dependent but static, and NRP, which is entirely data-independent. DPM provides an 88% improvement over the average breach count value exhibited by NRP respectively. For the hospital dataset, PCA exhibits a comparable performance due to the heavy correlation in the data.

   For synthetic data, our mechanism performs better than both the baselines, providing 12 and 10 times better performance than PCA and NRP, respectively. The poor performance of PCA is owing to the design of the mechanism which exploits the data correlation to find the important features to be retained.



(a)

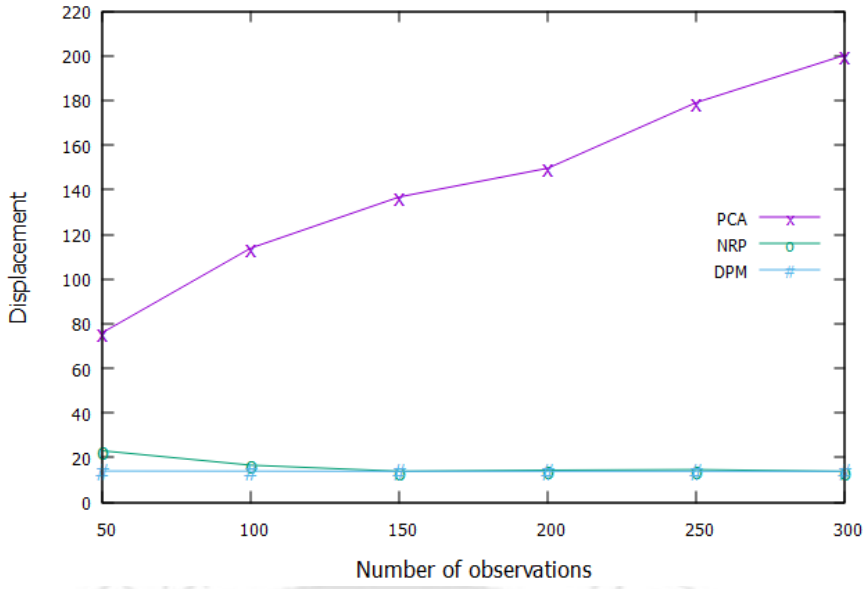


(b)

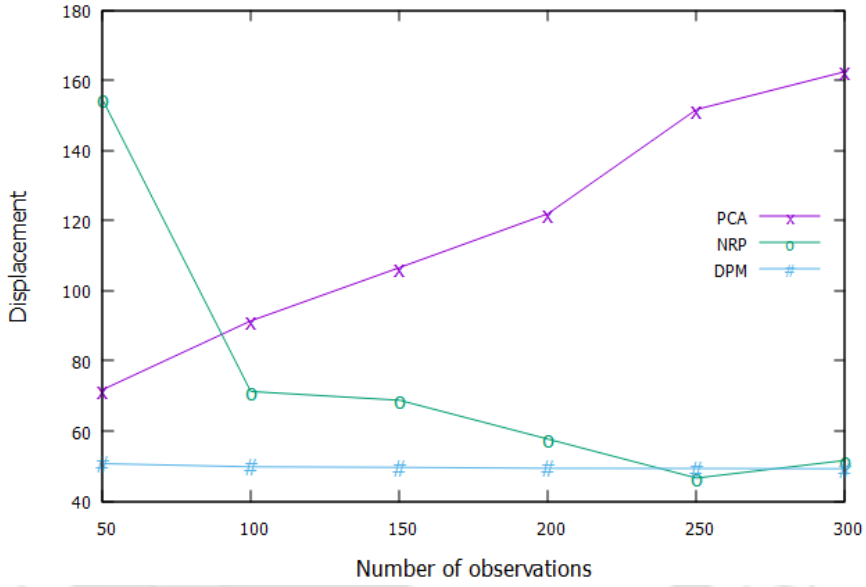
Figure 5.2: Breach Count Comparison for (a) Real life data (b) Synthetic data

2. Displacement analysis: As shown in Figure 5.3, our method yields a displacement value lower than those of NRP and PCA, indicating a balanced and stable utility in both datasets. For the real life dataset, the mean displacement values for NRP, DPM, and PCA are 16.0402, 13.8857, and 142.5848, respectively. In the synthetic data set, where the data points lack correlation, DPM still performs better than the baselines with an average displacement of 49.6891, compared to a significantly higher 117.6285 for PCA and 75.1016 for NRP.

3. Resemblance analysis: Figure 5.4 shows that all three methods produce



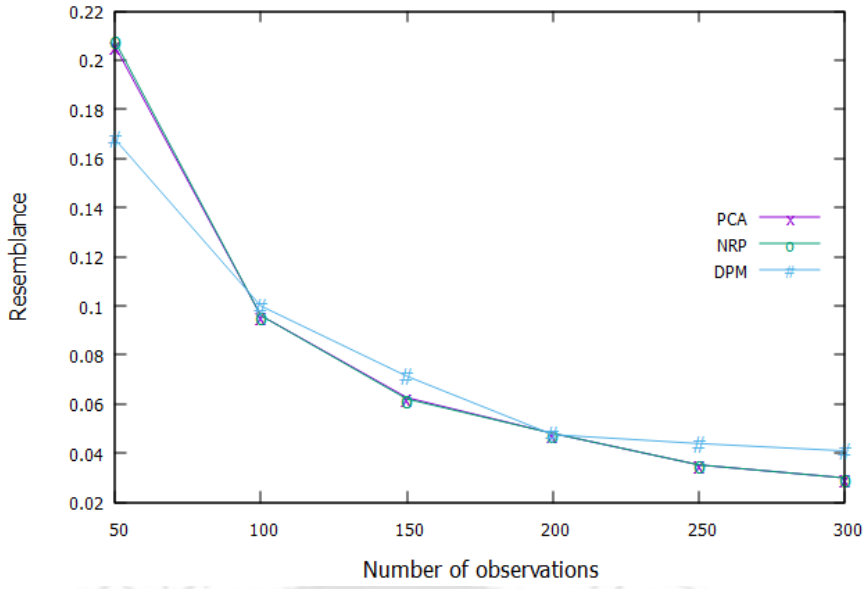
(a)



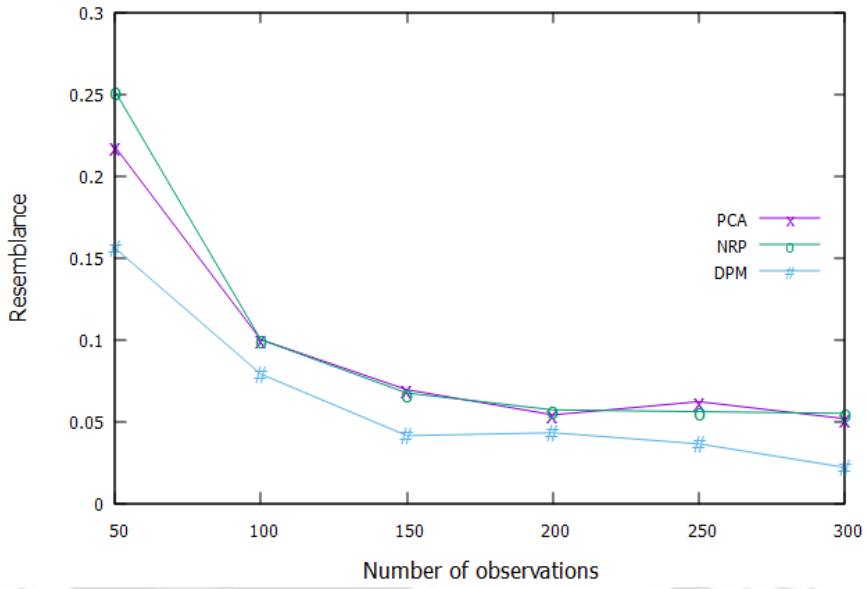
(b)

Figure 5.3: Displacement Comparison for (a) Real life data (b) Synthetic data

comparable resemblance values, consistent with the findings in [34] that dimension reduction techniques generally produce low resemblance. For the real life dataset, DPM achieves an average resemblance of 0.0786, offering a modest improvement over PCA (0.0796) and NRP (0.0798), which is approximately a 2% improvement over both the baselines. In the synthetic dataset, DPM performs significantly better, with an average resemblance of 0.0629, compared to 0.0925 for PCA and 0.0978 for NRP, representing improvements of about 32% and 35%, respectively.



(a)



(b)

Figure 5.4: Resemblance Comparison for (a) Real life data (b) Synthetic data

5.4.3 Discussion

PCA performs well in case of datasets with correlation, whereas NRP is data-independent and performs well for both sets of data. However, NRP explicitly incorporates a utility requirement, whereas our method, DPM, operates without the incorporation of any constraint on privacy or utility. Unlike NRP, our DPM approach is data dependent and outperforms NRP. Also, because of the in-place updation of the projection matrix, and the lack of complex preprocessing steps (as in PCA), DPM is computationally easier on the agents. Although PCA selectively achieves lower breach count and resemblance, it suffers from high displacement.

In contrast, DPM factors in previous observations, yielding a lower breach count, resemblance, and displacement, striking a better balance between privacy and utility overall. We present the formal computational analysis of the dimension reduction based methods in the next chapter (Section 6.5.1.2).

5.5 Conclusion

In this chapter, we studied projection based dimensionality reduction mechanisms for privacy preservation in MAS. We first introduced the problem of robust concept deduction and discussed how dimensionality reduction can help reduce overheads while preserving the ability of the fusion center to infer the underlying concept. As representative projection based mechanisms, we reviewed the BRP and NRP methods and discussed their characteristics.

BRP requires significant computational effort for the generation of an orthonormal matrix. NRP addresses this limitation by using random matrices with independently drawn bounded entries, which offers similar probabilistic guarantees. However, the per step matrix regeneration increases computational overhead and does not exploit information from previously observed data.

To address these limitations, we proposed DPM, a learning based projection mechanism that adaptively updates the projection matrix over time. The proposed approach generates the projection matrix once during initialization and subsequently updates it using the raw and sanitized data tuples observed at each step. The learning rate and regularization factor allow the mechanism to control the influence of new observations and maintain stability of the projection matrix, thereby helping balance privacy and utility without requiring additional constraints.

We provided theoretical analysis demonstrating that the proposed method preserves inner products with bounded distortion, thereby maintaining the geometric structure of the data and ensuring utility. Additionally, we showed that the cumulative privacy leakage remains bounded under differential privacy analysis, due to the controlled adaptive updates and periodic matrix refresh. Extensive experiments on both real life hospital data and synthetic datasets demonstrate that the proposed method achieves a favorable balance between privacy and utility. Overall, DPM provides an efficient and flexible framework for privacy preserving dimensionality reduction in decentralized environments such as MASs.

Chapter 6

Adaptive Modulo Hash Function

This chapter employs an adaptive modulo hash function for sanitization via dimension reduction. Traditional modulo hash functions rely on a single divisor for remainder calculation. Our proposed method solves a quadratic equation for the hash value, thereby mitigating attacks resulting from fixed divisor [1].

[1]The contents of this chapter appear will mostly in the paper titled "Preserving Inference Privacy in Multi-Agent Systems with Adaptive Modulo Hash Function" and partly in "Dimension Reduction via Random Projection for Privacy in Multi-Agent Systems", which are both currently under review.

Enforcing privacy using either noise addition or data degradation results in data distortion which in turn affects the utility of the data. Quantifying this loss is essential for evaluating the effectiveness of any privacy mechanism and for the enforcement of a constraint on utility or privacy. It enables system designers to find a tradeoff between utility and privacy. To this end, we use the cosine similarity between the raw and sanitized data vectors as a unified measure of both utility and privacy. Cosine similarity is invariant to scaling. This property ensures that, even if the sanitization process alters the magnitude of the data, the underlying directional information remains a reliable measure of utility preservation.

Building on the idea of dimension reduction for privacy preservation, we explore hashing-based mechanisms, namely Modulo hashing [63] for data sanitization. Its deterministic nature and high collision probability make it unsuitable for traditional cryptographic applications. However, these very characteristics can be leveraged for privacy preservation when appropriately modeled. Traditional modulo hashing requires the selection of a predefined divisor to compute the hash value. In this chapter, we propose an adaptive modulo hash function (AMHF) which does not require a fixed divisor and instead works by solving a quadratic equation for the hash value. The coefficients of the quadratic equation are formed using the parameters of the data tuple and the privacy requirement of the agent.

This chapter begins with the preliminaries, including the proposed definitions of utility and privacy and the motivation behind the AMHF algorithm in Section 6.1. This is followed by a brief recap of the system model in Section 5.1. We describe our proposed method in detail in Section 6.3, and the theoretical security evaluation in Section 6.4. The experimentation results are illustrated in Section 6.5.

6.1 Preliminaries

In this section, we introduce computationally efficient geometric measures of utility and privacy based on cosine similarity. These measures are intended to quantify the distortion introduced by sanitization while remaining simple to compute. We further establish their asymptotic relationship with the CRLB-based utility and privacy measures proposed in the literature under suitable regularity assumptions. Next, we highlight the issues of using the traditional modulo hashing method as a privacy preserving mechanism.

6.1.1 Quantification of Utility and Privacy

Cosine similarity has previously been used to measure the plagiarism between documents [62] and for face verification [55]. Its successful application in these

diverse domains motivates its use to quantify the loss in data utility introduced by sanitization. The similarity measure indicates how closely the sanitized data aligns with the original in terms of direction, with a higher cosine similarity implying greater similarity between the two vectors. The similarity between vectors $\mathbf{x}$ and $\mathbf{y}$, denoted as $cos(\mathbf{x}, \mathbf{y})$, is computed as given in Equation 6.1,

$$cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \times ||\mathbf{y}||}. \tag{6.1}$$

Computation of cosine similarity requires vectors of equal length. If the dimension of $\mathbf{y}$ is less than that of $\mathbf{x}$, the cosine similarity can be computed by embedding $\mathbf{y}_i$ back into the higher dimension (via zero padding). Since cosine similarity generally lies in the interval $[-1, 1]$, we impose the following assumption to ensure boundedness in $[0, 1]$. The sanitization mapping $\mathcal{M}$ preserves the sign of the inner product, i.e.,

$$\mathbf{y}_i^T \mathcal{M}(\mathbf{y}_i) \geq 0,$$

for all agents i. Under this assumption, the cosine similarity satisfies

$$0 \leq \cos(\mathbf{y}_i, \mathcal{M}(\mathbf{y}_i)) \leq 1. \tag{6.2}$$

The upper bound follows directly from the definition of cosine similarity. The lower bound is ensured by the nonnegativity of the inner product, which implies that $\mathbf{y}_i$ and $\mathcal{M}(\mathbf{y}_i)$ form an acute angle in the Euclidean space.

The proposed cosine-based privacy measure should be interpreted as a geometric measure of sanitization-induced distortion and as a computationally efficient surrogate for quantifying the privacy–utility tradeoff, rather than as a universal measure of information leakage. A lower Cosine similarity indicates greater geometric separation between the original and sanitized observations, which generally reduces their directional similarity. However, by itself, it does not preclude inference using auxiliary information, statistical side information, or application-specific knowledge. Consequently, throughout this chapter, $\mathcal{P}_{\cos}$ is employed as a computationally efficient design metric for controlling the privacy-utility tradeoff, while complementary security properties of the proposed mechanism are analyzed separately in Section 6.4.

Using the above results, we have the following definitions of utility and privacy:

Definition 6.1 *Utility of agent data: The utility of the data belonging to the agent i after application of the sanitization function $\mathcal{M}$ is given as $\mathcal{U}_{cos}(\mathcal{M}) = cos(\mathbf{y}_i, \mathcal{M}(\mathbf{y}_i))$.*

As the similarity between the tuples increases, the utility of the data increases. We say that the sanitized data provides perfect utility when the value of utility is

1.

Definition 6.2 *Agent privacy: The privacy achieved by agent i due to the application of the sanitization function $\mathcal{M}$ is given as $\mathcal{P}_{cos}(\mathcal{M}) = 1 - cos(\mathbf{y}_i, \mathcal{M}(\mathbf{y}_i)) = 1 - \mathcal{U}_{cos}(\mathcal{M})$.*

As the utility of the sanitized data increases, its geometric similarity to the original observation also increases, resulting in a lower value of the proposed cosine-based privacy measure. Since $cos(\mathbf{y}_i, \mathcal{M}(\mathbf{y}_i)) \in [0, 1]$, we subtract the similarity value from 1 to determine the level of privacy achieved.

We directly quantify $\mathcal{P}_{cos}(\mathcal{M}) = 1 - \mathcal{U}_{cos}(\mathcal{M})$ without the introduction of any other scaling parameter so that our formulation remains interpretable, and maps directly to the geometric notion of cosine similarity. The privacy of an agent can also be alternatively defined as $\mathcal{P}_{cos}(\mathcal{M}) = \alpha(1 - \mathcal{U}_{cos}(\mathcal{M}))$, with $\alpha > 0$, in cases where there is an asymmetry in the tradeoff. In many situations, modifying or disclosing certain aspects of the data can have a disproportionate effect on privacy compared to the resulting utility. That is, privacy and utility may respond to data changes at different rates, and their relationship may not be linearly inverse. In certain healthcare, finance, and social media scenarios, some parameters are far more privacy sensitive than others; even if those features significantly enhance utility, their disclosure may be unacceptable. In such cases, the parameter α provides flexibility and captures this asymmetry. However, the value of α is domain and design specific, requiring additional justification or empirical evidence. Without such evidence, introducing a scaling parameter could obscure the direct relationship between privacy and utility.

To validate our definitions of utility and privacy, we compare them with those proposed in the literature.

To theoretically justify the proposed geometric measures, we compare their asymptotic behaviour with the CRLB-based utility and privacy measures proposed by Wang et al. [69], which makes use of CRLB and FIM. Continuing with our earlier definitions of $\mathbf{x}$ as the system parameter and $\mathbf{y}$ as the agent's observation, we now examine the formulation proposed in Wang et al. [69]. In their framework, the CRLB for estimating $\mathbf{x}$ is defined as

$$\mathbf{P}_{\mathbf{x}} := (J_{\mathbf{x}} + J_0)^{-1}$$

where $J_{\mathbf{x}}$ and J_0 are Fisher information matrices. Taking $\mathcal{M}$ as the sanitization function, let $\tilde{\mathbf{y}} = \mathcal{M}(\mathbf{y})$ be the sanitized version of $\mathbf{y}$. The corresponding Fisher information is denoted as $J_{\mathcal{M}}$ and the updated CRLB is given by

$$\widetilde{\mathbf{P}}_{\mathbf{x}} := (J_{\mathcal{M}} + J_0)^{-1}.$$

For the asymptotic comparison between the two measures, we impose the following regularity conditions:

- The observation model is differentiable with respect to $\mathbf{x}$ and the corresponding FIM exists and is positive definite.
- The perturbation satisfies $\|\delta\mathbf{y}\| \ll \|\mathbf{y}\|$, so that second-order Taylor expansion is valid.
- The sanitization leads to a second-order reduction in Fisher information, that is, $J_{\mathcal{M}} = J_{\mathbf{x}} - \Delta$ with $\|\Delta\| \sim \mathcal{O}(\|\delta\mathbf{y}\|^2)$.
- Matrices U and H defining public and private parameter projections of the system parameter are fixed and of compatible dimensions.

Under these assumptions, in Theorem 6.1, we show that our proposed utility and privacy measures are bounded above by CRLB-based measures. The sanitized data is expressed as $\tilde{\mathbf{y}} = \mathbf{y} + \delta\mathbf{y}$ with $\delta\mathbf{y}$ denoting modification by compression or noise addition.

Theorem 6.1 *Let $\mathcal{P}_{\text{CRLB}}$ and $\mathcal{U}_{\text{CRLB}}$ be the privacy and utility values, respectively, computed using the CRLB framework. Similarly, let $\mathcal{P}_{\cos}$ and $\mathcal{U}_{\cos}$ denote the corresponding privacy and utility values computed using cosine similarity. Then under the assumption $\cos(\mathbf{y}, \tilde{\mathbf{y}}) \to 1$, there exist constants $C_1 > 0$ and $C_2 < 0$ such that:*

$$\begin{cases} \mathcal{P}_{\cos} \leq C_1 \cdot \mathcal{P}_{\text{CRLB}} + o(\mathcal{P}_{\text{CRLB}}), \\ \mathcal{U}_{\cos} \leq C_2 \cdot \mathcal{U}_{\text{CRLB}} + o(1). \end{cases} \tag{6.3}$$

The constants depend on the observation model, sanitization mechanism, and the availability of prior information.

Proof. We have $\delta\mathbf{y} := \tilde{\mathbf{y}} - \mathbf{y}$, and assume $\|\delta\mathbf{y}\| \ll \|\mathbf{y}\|$. By Taylor series expansion, we have:

$$\cos(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{\mathbf{y}^T(\mathbf{y} + \delta\mathbf{y})}{\|\mathbf{y}\| \cdot \|\mathbf{y} + \delta\mathbf{y}\|} \approx 1 - \frac{1}{2} \cdot \frac{\|\delta\mathbf{y}\|^2}{\|\mathbf{y}\|^2}.$$

Therefore:

$$1 - \cos(\mathbf{y}, \tilde{\mathbf{y}}) \approx \frac{1}{2} \cdot \frac{\|\delta\mathbf{y}\|^2}{\|\mathbf{y}\|^2}.$$

Data sanitization reduces Fisher information as follows:

$$J_{\mathcal{M}} = J_{\mathbf{x}} - \Delta, \quad \text{with } \|\Delta\| \sim \mathcal{O}(\|\delta\mathbf{y}\|^2).$$

This gives:

$$\begin{aligned}\widetilde{\mathbf{P}}_{\mathbf{x}} &= (J_{\mathcal{M}} + J_0)^{-1} \\ &= (J_{\mathbf{x}} + J_0 - \Delta)^{-1} \\ &\approx \mathbf{P}_{\mathbf{x}} + \mathbf{P}_{\mathbf{x}}\Delta\mathbf{P}_{\mathbf{x}} + o(\|\delta\mathbf{y}\|^2)\end{aligned}$$

Since U and H are the matrices specifying the public and private parameters, then:

$$\begin{aligned}\mathcal{P}_{\text{CRLB}} &\approx \frac{\text{Tr}(G\mathbf{P}_{\mathbf{x}}\Delta\mathbf{P}_{\mathbf{x}}G^T)}{\text{Tr}(G\mathbf{P}_{\mathbf{x}}G^T)} \sim \mathcal{O}(\|\delta\mathbf{y}\|^2) \\ \Rightarrow 1 - \cos(\mathbf{y}, \tilde{\mathbf{y}}) &\leq C_1 \cdot \mathcal{P}_{\text{CRLB}} + o(\mathcal{P}_{\text{CRLB}}) \\ \Rightarrow \mathcal{P}_{\cos} &\leq C_1 \cdot \mathcal{P}_{\text{CRLB}} + o(\mathcal{P}_{\text{CRLB}})\end{aligned}$$

Similarly, for utility:

$$\mathcal{U}_{\text{CRLB}} \approx -\frac{\text{Tr}(U\mathbf{P}_{\mathbf{x}}\Delta\mathbf{P}_{\mathbf{x}}U^T)}{\text{Tr}(U\mathbf{P}_{\mathbf{x}}U^T)}$$

Thus,

$$\begin{aligned}&|\mathcal{U}_{\text{CRLB}}| = \mathcal{O}(\|\delta\mathbf{y}\|^2), \text{ and } \mathcal{U}_{\text{CRLB}} \leq 0, \\ \Rightarrow &\mathcal{U}_{\cos} \leq C_2 \cdot \mathcal{U}_{\text{CRLB}} + o(1).\end{aligned}$$

□

In Theorem 6.1, the values of the constants C_1 and C_2 depend on multiple data and system specific characteristics. These include the values of $||\mathbf{y}||$ and the matrices $H, U, \mathbf{P}_{\mathbf{x}}$. That is, the constants depend on the data and are system specific. This theorem further establishes the asymptotic consistence up to second order in the perturbation magnitude.

CRLB-based methods require access to model-specific FIM and can be computationally intensive. In contrast, cosine similarity is straightforward to compute. The theorem suggests that our measures offer a practical and interpretable approximation to the CRLB-based metrics.

In this chapter, we incorporate a privacy constraint using our proposed definition of privacy to propose an adaptive hash based algorithm AMHF to enforce inference privacy, eliminating the need for a fixed division. We first motivate the elimination of a fixed divisor in the modulo hashing algorithm.

6.1.2 Motivation

The traditional modulo hash function computes the remainder of a division operation. In this approach, the numeric input data is divided by a predetermined

value, and the remainder serves as the hash value. This technique offers several advantages, making it a suitable choice for certain applications within the realm of privacy preservation. Mathematically, it can be represented as follows:

$$h(u) = r \equiv u \bmod d,$$

where:

- $h(u) = r$ represents the hash value of the input u which is the remainder when u is divided by d.
- d is the divisor.

The modulo hash function is computationally efficient, with a time complexity of $\mathcal{O}(1)$ for hash computation. This efficiency makes it suitable for application in MASs where fast processing of data is crucial for the system. The modulo hash function will always produce the same hash value $h(u)$ for a fixed u. The modulo hash function exhibits a uniform distribution of hash values across the output space.

Traditional modulo hashing poses security risks when the same divisor (d) is repeatedly used. This repetition creates patterns that could be exploited to decrypt hash values, compromising security. Now, we will briefly show that if we use the same divisor repeatedly, an adversary possessing the knowledge of multiple values of u, along with their hashes, can successfully compute the value of the divisor. This knowledge can be further used to compute the numeric value being hashed.

Let us use the same value of d for j-hash calculations, then

$$\begin{aligned} r_1 \equiv u_1 \bmod d &\implies u_1 - r_1 = k_1 d, \\ r_2 \equiv u_2 \bmod d &\implies u_2 - r_2 = k_2 d, \\ &\vdots \\ r_j \equiv u_j \bmod d &\implies u_j - r_j = k_j d. \end{aligned}$$

Now, by prime factorization, we get,

$$\begin{aligned} k_1 d &= \ell_1^{a_{1,1}} \times \ell_2^{a_{1,2}} \times \cdots \times \ell_i^{a_{1,i}}, \\ k_2 d &= \ell_1^{a_{2,1}} \times \ell_2^{a_{2,2}} \times \cdots \times \ell_i^{a_{2,i}}, \\ &\vdots \\ k_j d &= \ell_1^{a_{j,1}} \times \ell_2^{a_{j,2}} \times \cdots \times \ell_i^{a_{j,i}}. \end{aligned}$$

No. of common divisors =

$$[(a_{1,1}, a_{2,1}, \ldots, a_{j,1}) + 1][(a_{1,2}, a_{2,2}, \ldots, a_{j,2}) + 1] \cdots [(a_{1,i}, a_{2,i}, \ldots, a_{j,i}) + 1]. \tag{6.4}$$

Here, $(x, y, \cdots, z)$ denotes the minimum among $x, y, \cdots, z$ and $\ell_1, \cdots, \ell_i$ are the prime divisors.

$\implies$ d will be one of these divisors. Now if the adversary makes one more appropriate query then they will have the value of d. If the greatest common divisor of any two $k_j d$'s is a prime, the value of d can be deduced with the knowledge of as few as two hash values.

Example 6.2 *Let the value of the divisor (d) be 10 which the adversary does not know. Let us assume the adversary issues the queries:* 11 *and* 17 *respectively. Then,*

$$h(11) = 11 \ mod \ 10 = 1 \implies 11 - 1 = 10 = 1 \times 10 = 2^1 \times 5^1,$$

$$h(17) = 17 \ mod \ 10 = 7 \implies 17 - 7 = 10 = 1 \times 10 = 2^1 \times 5^1.$$

Now, $gcd(10, 10) = 10 = 2^1 \times 5^1$. *Thus, the number of common divisors and subsequently, the number of choices for* $d = 4$, *namely,* $1, 2, 5$ *and* 10. *The agent now needs to make an informed query, which has to be a number that would return a different hash value for each of the divisors. In this case, a possible query value can be 26. Now,*

$$h(26) = 26 \ mod \ 10 = 6 \implies d = 10.$$

To address this vulnerability, this thesis proposes a novel hashing methodology that dynamically adjusts the value of the divisor, thereby enhancing hash function unpredictability and strength. The method would also incorporate the privacy requirement (ϵ) of the agent during the hash value calculation. Using this method, the agents would sanitize the data in a decentralized manner according to their privacy requirement. The sanitized tuple is then sent to the fusion center.

6.2 System Model

We consider an S agent MAS model as introduced in Section 1.2. We illustrate our methodology for a single time instant. Therefore, without loss of generality, we assume the observation of agent i, that is, $\mathbf{y}_i$ to be of the form

$$\mathbf{y}_i = (p_1, p_2, \ldots, p_k, g_1, g_2, \ldots, g_{n-k}), \tag{6.5}$$

where,

p_i's are the private parameters $\forall i$ such that $1 \leq i \leq k$, and g_j's are the public parameters $\forall j$ such that $1 \leq j \leq n-k$.

Each agent has a specific requirement of privacy ($\epsilon \in (0,1)$) which is set according to the privacy-utility tradeoff admissible for the agent, which the agent specifies during sanitization. Our approach is that each agent will employ a sanitization process involving an adaptive modulo hash function h, differing solely in their privacy requirement $\epsilon \leq \mathcal{P}_{cos}(\mathcal{M}) \leq 1$. The hash function is applied to the private data, generating a secure hash output that is transmitted along with the public data to the fusion center. In the next section, we illustrate our proposed methodology, which helps each agent attain its individually prespecified level of privacy (ϵ).

6.3 Proposed Methodology

We propose AMHF, which is to be applied by each agent in a decentralized manner on the data tuple before sending it to the fusion center. The proposed method avoids the selection of a fixed value for d. The hash value is computed using the data tuple elements and the privacy parameter ϵ, which defines the agent's privacy requirement. We illustrate the method for a single agent. Each agent will follow a similar method differing only in the value of ϵ. Let the observation tuple $\mathbf{y} \in \mathbb{R}^n$ be structured as given in Equation 6.5.

Now, we will use the adaptive modulus hash function to sanitize our observation tuple $\mathbf{y}$. To enable message authentication, we communicate one private parameter without modification. Without loss of generality, let this parameter be p_1. Let X be the number obtained by concatenating $p_2, p_3, \ldots, p_k$, *i.e.*,

$$X = p_2 p_3 \ldots p_k.$$

The hash function h is then defined as:

$$h : X \bmod d = r,$$

where r is the hash value. The data communicated by the agent to the fusion center will thus be of the following form:

$$\mathcal{M}(\mathbf{y}) = (p_1, h(X), g_1, g_2, \ldots, g_{n-k}),$$

that is,

$$\mathcal{M}(\mathbf{y}) = (p_1, r, g_1, g_2, \ldots, g_{n-k}) \in \mathbb{R}^m, \tag{6.6}$$

where $m = n - k + 2$.

However, the calculation of the hash value must also incorporate the privacy requirement (ϵ) of the agent. When an agent sanitizes the data in a decentralized way, it will incorporate its privacy requirement. The sanitization function is based on the following theorem.

Theorem 6.3 *Given a privacy requirement $\epsilon \in (0, 1)$, the remainder r of the AMHF can be computed by solving a quadratic equation whose coefficients are functions of ϵ, p_i's and g_i's that is, the privacy requirement and the private and public parameters respectively.*

Proof. From the definition of privacy, we have:

$$\epsilon \leq 1 - cos(\mathbf{y}, \mathcal{M}(\mathbf{y})) \leq 1,$$

which, upon simplification, gives

$$0 \leq \frac{\mathbf{y} \cdot \mathcal{M}(\mathbf{y})}{||\mathbf{y}||_2 ||\mathcal{M}(\mathbf{y})||_2} \leq \beta,$$

where $\beta = 1 - \epsilon$. The sanitized tuple $\mathcal{M}(\mathbf{y}) \in \mathbb{R}^m$ can be written in the following form:

$$\mathcal{M}(\mathbf{y}) = (p_1, r, 0, 0, \ldots, 0, g_1, g_2, \ldots, g_{n-k}) \in \mathbb{R}^n.$$

Public parameters are kept intact because we are dealing with inference privacy only. Calculating $\mathbf{y} \cdot \mathcal{M}(\mathbf{y})$, $||\mathbf{y}||_2$ and $||\mathcal{M}(\mathbf{y})||_2$ in terms of $p_1, p_2, \ldots, p_k, g_1, g_2, \ldots, g_{n-k}$ and r, we have

$$\frac{\mathbf{y} \cdot \mathcal{M}(\mathbf{y})}{||\mathbf{y}||_2 ||\mathcal{M}(\mathbf{y})||_2} = \frac{p_1^2 + p_2 r + z_1}{\sqrt{z_1 + z_2}\sqrt{p_1^2 + r^2 + z_1}} \leq \beta,$$

where,

$$z_1 = g_1^2 + g_2^2 + \ldots + g_{n-k}^2,$$

and,

$$z_2 = p_1^2 + p_2^2 + \ldots + p_k^2.$$

Squaring both sides and grouping like terms, we have the following equation:

$$ar^2 + br + c \leq 0, \tag{6.7}$$

where,

$$\begin{cases} a = p_2^2 - \beta^2(z_1 + z_2), \\ b = 2p_2(p_1^2 + z_1), \\ c = (p_1^2 + z_1)(p_1^2 + z_1 - \beta^2(z_1 + z_2)). \end{cases} \tag{6.8}$$

□

The solution of Equation 6.7 will give the required hash value. We will now show that Equation 6.7 always has a solution. Equation 6.7 does not have a solution only when both of the following conditions hold together:

$$a > 0, \tag{Condition 1}$$

and

$$b^2 - 4ac < 0. \tag{Condition 2}$$

Algorithm 4 AMHF sanitization algorithm for i-th agent

Require: *Data tuple* $\mathbf{y}_i$ *of length* n, *Privacy requirement* ϵ

Ensure: *Data tuple* $\mathcal{M}(\mathbf{y}_i)$ *of dimension* m

$\beta \leftarrow 1 - \epsilon$
Calculate z_1, z_2
$a \leftarrow p_2^2 - \beta^2(z_1 + z_2)$
$b \leftarrow 2p_2(p_1^2 + z_1)$
$c \leftarrow (p_1^2 + z_1)(p_1^2 + z_1 - \beta^2(z_1 + z_2))$
if $(b^2 - 4ac) < 0$ **then**
    $r \leftarrow 1$
else
    $root1 \leftarrow (-b + \sqrt{b^2 - 4ac})/(2a)$
    $root2 \leftarrow (-b - \sqrt{b^2 - 4ac})/(2a)$
    $final\_root \leftarrow \max(root1, root2)$
    $r \leftarrow \lceil final\_root \rceil$
    if $a > 0$ **then**
        $r \leftarrow r - 1$
    else
        $r \leftarrow r + 1$
    end if
end if
$\mathcal{M}(\mathbf{y}_i) = (p_1, r, g_1, g_2, \ldots, g_{n-k})$

We will now show that Condition 1 and Condition 2 are not possible together.

Rewriting $b^2 - 4ac$ using Equation 6.8, we have

$$\begin{aligned} & b^2 - 4ac \\ & = 4\beta^2(p_1^2 + z_1)(z_1 + z_2)(p_1^2 + p_2^2 + z_1 - \beta^2(z_1 + z_2)). \end{aligned}$$

All the terms appearing in $b^2 - 4ac$ are positive as they are either the square or sum of squares of real values. The only term which can take a negative value is $p_1^2 + p_2^2 + z_1 - \beta^2(z_1 + z_2)$, which can be written as $p_1^2 + z_1 + a$.

Now if, $a = p_2^2 - \beta^2(z_1 + z_2) > 0$, then,

$$b^2 - 4ac > 0.$$

Hence, a solution of Equation 6.7 always exists.

Our proposed AMHF algorithm which will be applied by each agent in a decentralized manner is outlined in Algorithm 4. For the i-th agent, the input to the algorithm is the data tuple $\mathbf{y}_i$ and the privacy requirement ϵ. The algorithm then computes the solution of the quadratic equation given in Equation 6.7 formed using the tuple elements. We choose an appropriate solution r depending on the discriminant of the quadratic equation. The output of the algorithm is the sanitized data $\mathcal{M}(\mathbf{y}_i)$, which is of the form given in Equation 6.6. When two distinct real roots exist, the larger root is selected, followed by the rounding step described in Algorithm 4, to ensure that the selected integer value lies within the admissible region of the quadratic inequality. If the quadratic has a repeated real root, the two roots coincide and the same selection rule is applied. Complex roots arise only when the discriminant is negative. Therefore, the algorithm assigns the default value $r = 1$ as a fallback, guaranteeing that the sanitization procedure always produces a valid output. Any other predetermined integer could equally be used; the choice of 1 is made solely for simplicity and reproducibility.

We now illustrate the working of our mechanism using Example 6.4.

Example 6.4 *For a particular agent, let the observation tuple be* $\mathbf{y} = (5, 37, 84, 55, 81, 65, 13, 86) \in \mathbb{R}^8$ *and the privacy requirement* $\epsilon = 0.5$. *The first* 3 *tuple elements are assumed to be private parameters while the rest are public. Here,* $p_1 = 5$, $p_2 = 37$, $p_3 = 84$. *This gives,*

$$z_1 = g_1^2 + g_2^2 + \ldots + g_5^2 = 21376,$$

$$z_2 = p_1^2 + p_2^2 + p_3^2 = 8450.$$

Then we have the following values of a, b *and* c*:*

$$a = p_2^2 - \beta^2(z_1 + z_2) = -6087.5,$$

$$b = 2p_2(p_1^2 + z_1) = 1583674,$$
$$c = (p_1^2 + z_1)(p_1^2 + z_1 - \beta^2(z_1 + z_2))$$
$$= 298426244.5.$$

Thus the quadratic equation is:

$$-6087.5r^2 + 1583674r + 298426244.5 \leq 0, \tag{6.9}$$

or,

$$-r^2 + 260.15r + 49022.79 \leq 0. \tag{6.10}$$

The two roots are -126.72 *and* 386.87. *The maximum of the two roots is* 386.87. *Since the value of a is less than zero, the hash value* $h(X) = \lceil 386.87 \rceil + 1 = 388$. *Therefore the sanitized tuple is,*

$$\mathcal{M}(\mathbf{y}) = (5, 388, 55, 81, 65, 13, 86) \in \mathbb{R}^7.$$

6.4 Security Evaluation

In this section, we carry out a thorough security evaluation of our mechanism. We first show that our hash function is not homomorphic. That is, the hash value of the sum of the inputs is not the same as the sum of the individual hash values of the inputs. Next, we prove that our method leaks at most a small ϵ-dependent amount of information about the sensitive parameters. We have the following two corollaries as part of the security evaluation.

Corollary 6.5 *The proposed AMHF algorithm is not homomorphic.*

Proof. Since our algorithm computes the value of hash as a root of a quadratic equation, AMHF is not homomorphic. Let $\mathbf{y_1}$ and $\mathbf{y_2}$ be any two readings from the same agent. Let us also assume that the privacy requirement of the agent stays the same. The coefficients a, b and c are constructed using z_1 and z_2. Since the sum of squares is not equal to the square of sums, the values of the coefficients make our algorithm non-homomorphic.

We will further substantiate our claim that our method is not homomorphic by giving an example.

Example 6.6 *Let* $\mathbf{y}_1 = (1, 2, 3, 7, 8, 9)$ *and* $\mathbf{y}_2 = (4, 5, 6, 7, 8, 9)$ *be two readings from a single agent with fixed* $\epsilon = 0.5$. *Then from calculations,* $\mathcal{M}(\mathbf{y}_1) = (1, 135, 7, 8, 9)$ *and* $\mathcal{M}(\mathbf{y}_2) = (4, 171, 7, 8, 9)$. *However,*

$$\mathcal{M}(\mathbf{y}_1 + \mathbf{y}_2) = (5, 198, 14, 16, 18) \neq \mathcal{M}(\mathbf{y}_1) + \mathcal{M}(\mathbf{y}_2).$$

□

This implies that our method is immune to computation on the hashed private parameters. Patterns cannot be exploited to breach the privacy of the agent. It is also immune to tampering.

Corollary 6.7 *The proposed AMHF leaks at most a small ϵ-dependent amount of information about the sensitive parameters.*

Proof. For the proof, we use mutual information as the privacy metric. It is denoted as $MI(a;b)$ and quantifies the number of bits of information about a obtained from knowing b. We wish to obtain a bound on $MI(X;r)$. It can be written as

$$\begin{aligned} &MI(X;r) \\ &= E(r|p_1, g_1, \cdots, g_{n-k}) - E(r|X, p_1, g_1, \cdots, g_{n-k}), \end{aligned}$$

where $E(\cdot|\cdot)$ gives the conditional entropy. Since our method is deterministic,

$$E(r|X, p_1, g_1, \cdots, g_{n-k}) = 0.$$

Thus,

$$MI(X;r) = E(r|p_1, g_1, \cdots, g_{n-k}). \tag{6.11}$$

The design of the mechanism is such that many different values of X get mapped to the same hash value r. Formally, we define the preimage set for any r as

$$\mathcal{X}_r = \{X : h(X) = r\}.$$

Collisions are essential for preserving inference privacy. A fixed combination of private and public parameters will always yield the same result, but our method is such that multiple combinations of $p_2, p_3, \ldots, p_k$ yield the same hash value. This ensures that an adversary cannot uniquely determine the preimage from the output of AMHF. A larger cardinality of $\mathcal{X}_r$ implies enhanced difficulty in inferring the exact preimage of r. By a careful choice of ϵ, we ensure that the cardinality of the range of r, that is, $\mathcal{R}$ is bounded. We assume that

$$|\mathcal{R}| \leq K(\epsilon),$$

where $K(\epsilon)$ is a function decreasing in ϵ. Then the mutual information is

$$\begin{aligned} MI(X;r) &= E(r|p_1, g_1, \cdots, g_{n-k}) \\ &\leq \ln(K(\epsilon)). \end{aligned} \tag{6.12}$$

Considering $K(\epsilon)$ to be such that $\ln(K(\epsilon)) \leq \epsilon$, we have the guarantee that the output r reveals at most ϵ bits of information about the input X. □

6.5 Experimentation and Results

The adversary's objective is to infer the private parameters of the agent by analyzing the sanitized data. This section aims to characterize the potential strategies and capabilities of the adversary in exploiting vulnerabilities within the data sanitization process. We consider two broad types of attacks, namely, data reconstruction and brute-force attack.

Consider that the agent A_i has the following raw data: $(p_1, p_2, \cdots, p_k, g_1, g_2, \cdots, g_{n-k})$. The sanitized data after hashing is $(p_1, r, g_1, g_2, \cdots, g_{n-k})$. For data security, we may additionally implement a cryptographic mechanism, namely an asymmetric key mechanism (key established by a handshake).

Adversary model: We consider an active adversary who wishes to gain knowledge of the private parameters of the agents from the sanitized data. The primary objective of the adversary is to deduce $p_2, p_3, \ldots, p_k$ from the sanitized data $(p_1, r, g_1, g_2, \ldots, g_{n-k})$. Now X is formed by concatenating $p_2, p_3, \cdots, p_k$. Let each p_i have a pre-specified format, that is, let each p_i $(i = 1, 2, \cdots, k)$ have a q-digit format. Thus the number X is a $q(k-1)$ digit element. The adversary is assumed to know the complete AMHF algorithm, the sanitization procedure, all public parameters, and the transmitted sanitized tuple. The unknown quantities are the private parameters $(p_2, \ldots, p_k)$ (equivalently X). Consequently, the security of AMHF does not rely on secrecy of the algorithm, but on the computational difficulty of recovering the private parameters from the observed hash value.

6.5.1 Data Reconstruction Attacks - Numerical results

In this section, we consider that the adversary uses a reconstruction based method to get back the original data. Since our method is a dimension reduction based method, it can also be thought of as a projection of the data tuple to a lower dimensional space. The attacker can use a suitable inverse-projection method to reconstruct the original data tuple. For our experimentation, we have considered a matrix based reconstruction algorithm wherein the attacker generates an inverse matrix to obtain a reconstruction of the original datapoint.

6.5.1.1 Algorithms used for comparison

We have initially compared our algorithm with other existing dimension reduction mechanisms in terms of computational complexity. In particular, we compare our algorithm with PCA [48], BRP [67], NRP [34] and DPM [Chapter 5]. Further, we experimentally compare AMHF to NRP and DPM.

6.5.1.2 Complexity comparison

In this section, we compare the computational complexity of our algorithm with that of the other dimensionality reduction based privacy preserving methods, namely PCA, BRP, NRP and DPM. We subdivide the complexity into two parts based on the step in which it is incurred: the preprocessing and sanitization steps. The complexity values represent arithmetic-operation counts under the standard RAM computational model, which is commonly used for asymptotic algorithm analysis.

The complexities are tabulated in Table 6.1.

Algorithm	Pre-processing	Sanitization
PCA	$\mathcal{O}(Nn^2 + n^3)$	$\mathcal{O}(nm)$
BRP	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$
NRP	$\mathcal{O}(nm)$	$\mathcal{O}(nm)$
DPM	Initialization: $\mathcal{O}(nm)$ Re-orthogonalization: $\mathcal{O}(nm^2/T_r)$ Refresh: $\mathcal{O}(nm/T_f)$	Projection: $\mathcal{O}(nm)$ Update: $\mathcal{O}(nm)$
AMHF	–	$\mathcal{O}(n)$

Table 6.1: Complexity comparison.

PCA involves a computationally intensive preprocessing step where the components of highest variance are identified by the computation of the eigen values and the eigen vectors. BRP involves the generation of a orthonormal matrix as part of the preprocessing. The projection matrix updation step involves matrix multiplication step which incurs a $\mathcal{O}(nm)$ complexity. On the other hand, NRP generates a fresh random matrix at each sanitization step, thereby eliminating the need of any preprocessing. However, the complexity of sanitization for all the dimension reduction methods mentioned thus far, is $\mathcal{O}(nm)$ as all of them involve matrix multiplication for projection of the raw tuple onto a lower dimensional subspace.

The proposed AMHF algorithm computes the hash value by solving a quadratic equation. Computing the values of the coefficients of the quadratic equation has a

complexity of $\mathcal{O}(n)$ for an n-tuple raw data. The remaining steps contribute only constant computational cost. Thus, our proposed method is by far the most efficient mechanism in comparison. This gives AMHF an edge over the other methods in its applicability in MAS models, where the agents have limited computational resources.

6.5.1.3 Datasets used

- Synthetic data: We consider that an agent makes multiple observations, varying from 1 to 300. The observation tuple of the agent $\mathbf{y}_i$ is of length 50, out of which 12 parameters are private while the rest are public. The values in the data tuple are independently drawn from the uniform distribution, $\mathrm{U}(-0.5, 0.5)$.

	AGE	GENDER	.....	ALCOHOL	HB	TLC	PLATELETS	GLUCOSE	UREA	.....	BNP
0	81	1	.....	0	9.5	16.1	337	80	34	.....	1880
1	60	0	.....	0	13.6	9.1	26	144	55	.....	1840
2	44	1	.....	0	13.5	22.3	322	217	51	.....	1720
3	56	0	.....	0	13.3	12.6	166	277	28	.....	518
4	52	1	.....	0	13.2	7.9	227	263	45	.....	780
5	54	1	.....	0	14.7	3.5	280	92	180	.....	23
6	85	1	.....	0	12.7	12	308	180	28	.....	823
7	56	1	.....	0	12	8.2	332	122	39	.....	2160
8	65	1	.....	0	10	10.1	457	152	16	.....	1780
9	58	0	.....	0	9.5	12.5	258	261	118	.....	1930
10	63	1	.....	0	14.3	10.1	224	141	63	.....	432

Figure 6.1: Hospital Dataset.

- Real life hospital dataset: This data was collected from patients admitted over a period of two years[9]. A snippet of the dataset is shown in Figure 6.1. We have dropped a few non-numeric data fields, and converted binary non-numeric data fields (such as gender) to binary numeric data fields. Each row of the dataset has multiple parameters of a single patient ranging from identification information to health data. We have considered identification parameters like age, gender, type of geographic area of residency to be the private parameters while the health records like HB, TLC, platelets are public. For each patient, we have a 50-tuple observation data out of which 14 are considered to be private parameters while the rest are public parameters.

6.5.1.4 Performance metrics

For comparing the performance of the mechanisms with respect to utility and privacy, we have used the metrics listed in Section 5.4.1.1 but interpret them differently.

- Breach Count [34]: This metric represents the average number of data points that are accurately reconstructed within a specified neighborhood of their original values.

  For the computation of breach count we have taken a threshold of 0.1. That is, for a breach, the reconstructed tuple has to lie within a $\pm 10\%$ neighborhood of the raw tuple. For example, if the raw data tuple is $(1, 2, 3, 4, 5)$ and the reconstructed data point is $(1.05, 1.95, 3, 4, 5)$, then we have a breach. We repeat the reconstruction algorithm 100 times, and take the average of the breach count. A lower breach count indicates a higher level of privacy preservation.

- Resemblance [34]: This metric quantifies the average number of neighbors that are common between each original data point and its corresponding reconstructed version. A lower resemblance value suggests better privacy preservation, as it indicates that the reconstructed data does not closely mimic the neighborhood structure of the original data.

- Displacement [34]: Displacement measures the average Euclidean distance between original and reconstructed data points across all agent observations. It is expressed in standard Euclidean distance units. Higher values of displacement implies higher levels of achieved privacy, and lower levels of data utility.

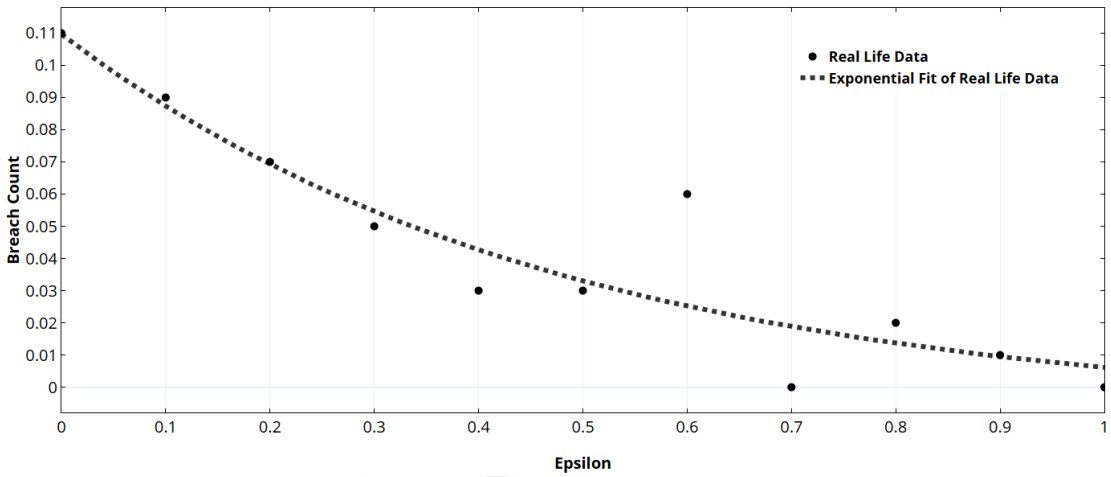
6.5.1.5 Variation of Breach Count with respect to epsilon (ϵ) for a single observation

For a single agent observation, we have found the breach count for varying levels of privacy (ϵ). As seen from Figure 6.2, breach count decreases exponentially as the value of ϵ increases. This result holds for both real life as well as synthetic data. The experimental results are in accordance with the theoretical results, wherein the level of privacy increases with an increase in the value of ϵ.

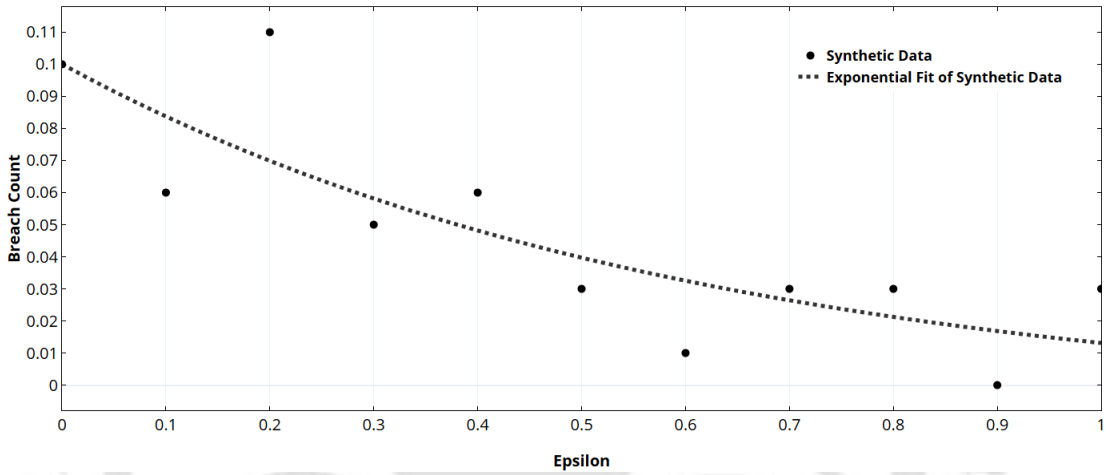
6.5.1.6 Results with varying number of observations

The breach count, displacement and resemblance results are plotted in Figure 6.3-6.5 respectively. The results illustrated in the section have been obtained with $\epsilon = 0.6$. Other values of ϵ also yielded similar trends.

1. Breach Count analysis: From Figure 6.3, we see that for the real dataset, AMHF demonstrates a consistent reduction in breach count as the number of observations grows, decreasing from a relatively higher value at smaller sample sizes to nearly zero at 300 observations. This indicates that AMHF



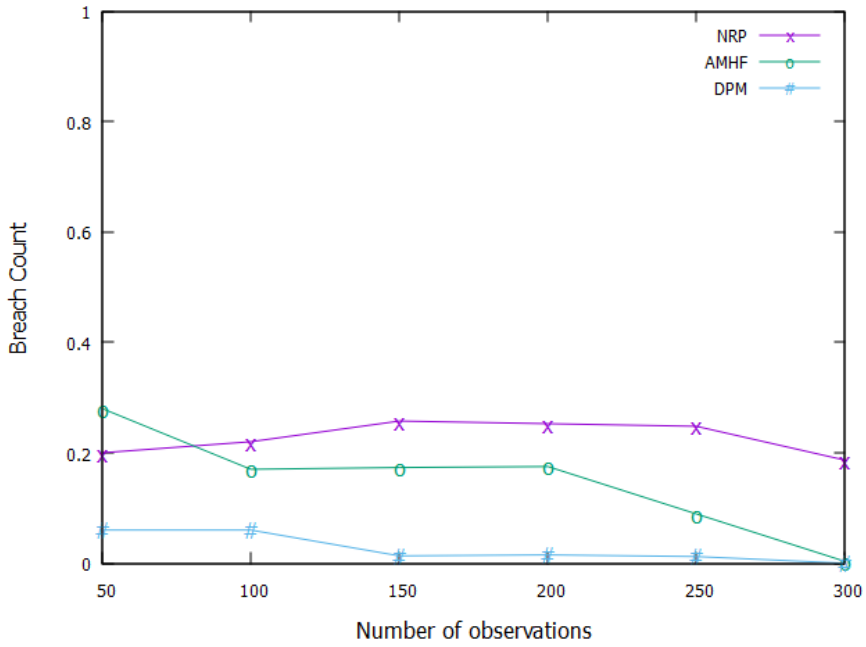
(a)



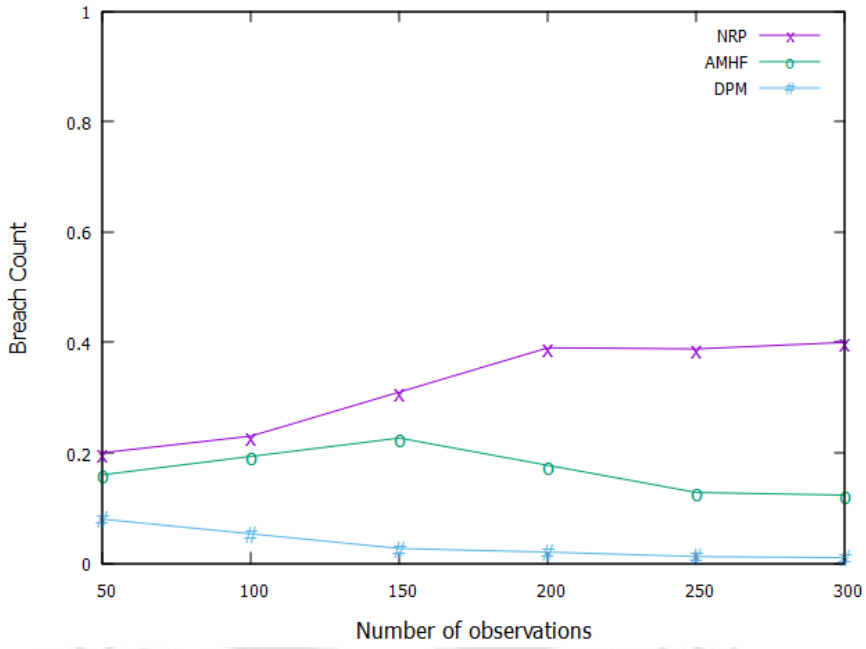
(b)

Figure 6.2: Variation of Breach Count with respect to ϵ (a) Real life data (b) Synthetic data.

becomes increasingly reliable as more data are available. This behavior can be attributed to the adaptive nature of the AMHF mechanism. As the number of observations increases, the quantities z_1 and z_2, representing the squared norm components of the data tuple, become more stable. This results in more consistent quadratic coefficients and, consequently, more reliable hash values that satisfy the imposed privacy constraint. Consequently, the likelihood of privacy violations decrease. On the other hand, NRP maintains a relatively high and fluctuating breach count across most observation levels. This is because the projection matrix at each step is generated randomly, independent of the underlying data distribution. DPM consistently produces the lowest breach counts overall, though AMHF approaches similar performance at larger observation sizes. The performance of DPM is due to its design which uses the data at each step for matrix updation. This high



(a)



(b)

Figure 6.3: Breach Count Comparison for (a) Real life data (b) Synthetic data.

data dependency of DPM ensures that the distribution of the data plays a major role in privacy retention.

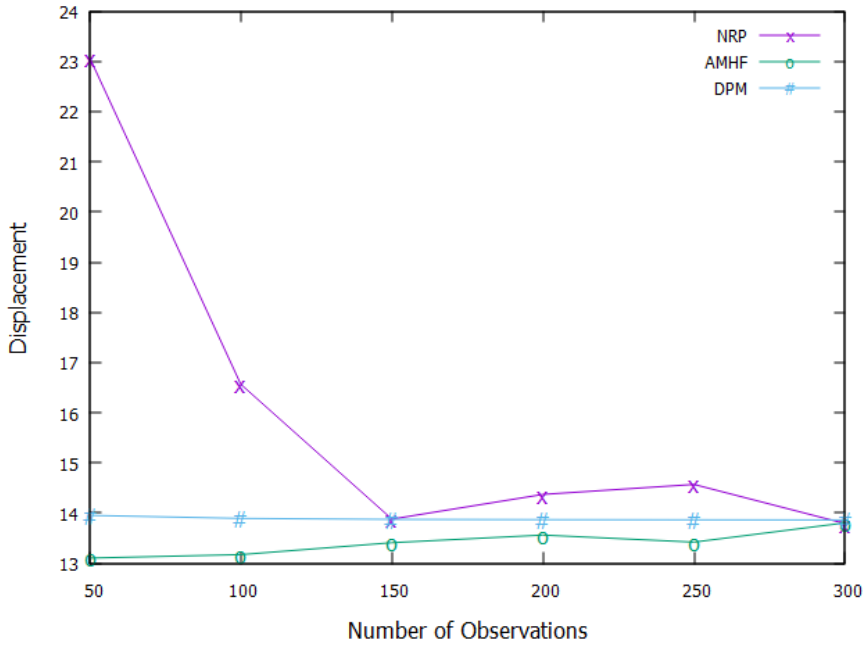
In the synthetic dataset, correlation between two data points is minimal. AMHF initially shows a moderate breach count that peaks around 150 observations before steadily declining as observations increase. This behavior arises because the hash value in AMHF depends on the tuple elements, and

with weakly correlated data the early values of the parameters z_1 and z_2 lead to occasional violations of the privacy bound. However, as more observations become available, the computed hash values stabilize leading to AMHF maintaining a controlled and generally decreasing breach rate. This demonstrates improved stability relative to NRP though still slightly weaker than DPM. However, owing to the linear computational complexity of AMHF, it is more efficient than DPM and NRP which involve computationally heavier matrix multiplications.

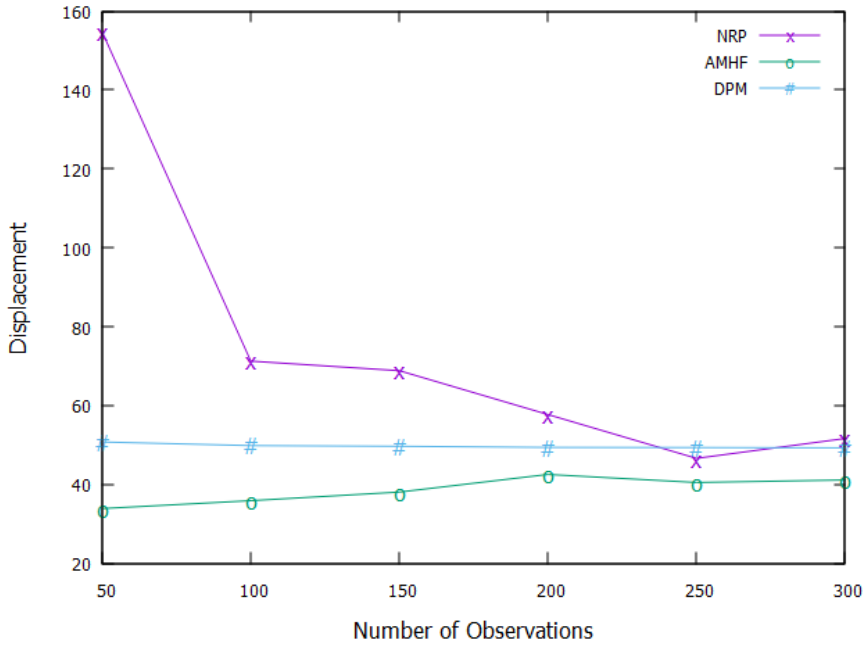
2. Displacement analysis: In case of real life data, from Figure 6.4, we see that our algorithm fares better than NRP and is comparable to DPM. This is because both AMHF and DPM are data dependent. While AMHF uses the tuple elements to form the quadratic equation, DPM uses the data tuples to update the projection matrix. This explains the relatively stable displacement values for AMHF and DPM. Similarly, for synthetic data too, the performance of our mechanism is comparable to DPM. The data independent NRP provides higher values for displacement in both cases. This implies that our method also preserves utility comparable to DPM and better than NRP, even while handling weakly correlated data. However, our proposed AMHF algorithm does not involve matrix multiplication based preprocessing or sanitization step, and only involves the computation of the sum of the squares of the tuple elements. This computation incurs a $\mathcal{O}(n)$ complexity, which is lighter on the resource constrained agents as compared to the matrix multiplication based methods like DPM and NRP.

3. Resemblance analysis: From Figure 6.5, we see that our algorithm fares equally well as compared to the other three algorithms for both real-life as well as synthetic data. This is consistent with findings in the literature [34], where dimension reduction based methods have similar resemblance values, lower than that exhibited by noise-addition based methods. The use of a different matrix at each step in the NRP and DPM mechanisms, and the use of a quadratic equation to find the hash value enhances the randomness in the sanitized result. Subsequently, the structural similarity between the raw and reconstructed datapoints weakens, leading to a reduction in the number of common neighbours between the raw and reconstructed datapoint, and therefore a reduced value of resemblance.

Discussion: The average metric values (with the standard deviation over 100 runs as the variability measure) are summarized in Table 6.2. NRP method provides better values for displacement for both the datasets. However, owing to the data independence, the breach count and resemblance is comparatively higher



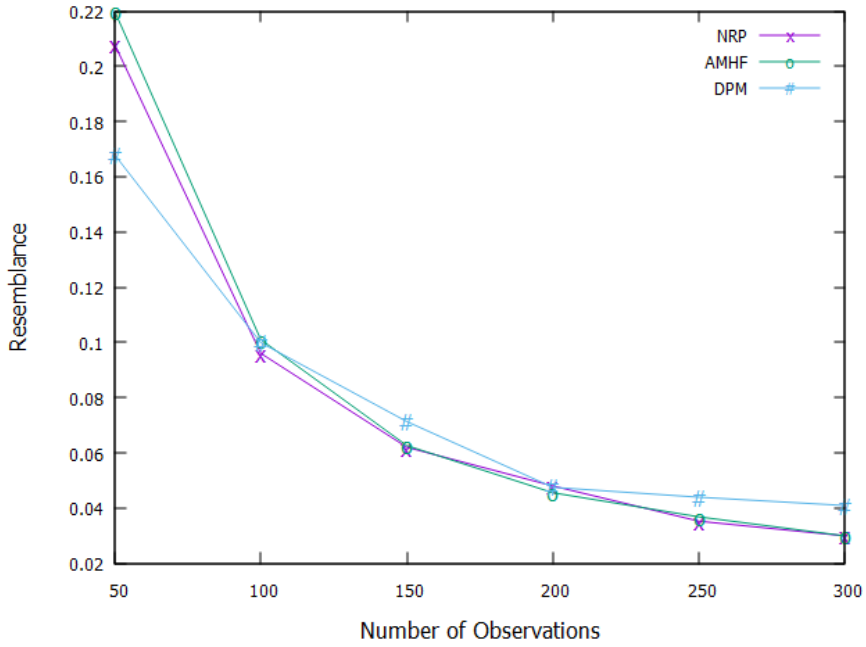
(a)



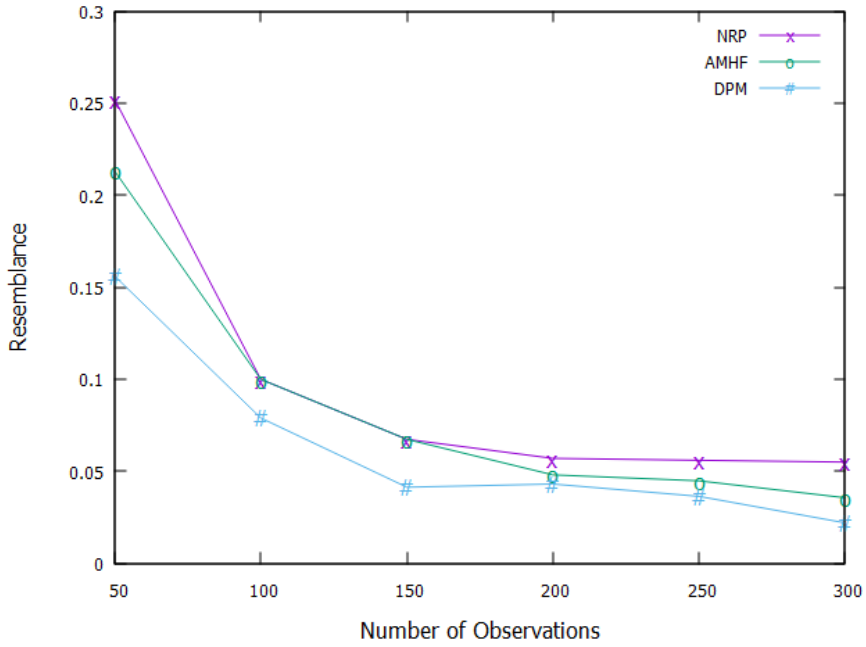
(b)

Figure 6.4: Displacement Comparison for (a) Real life data (b) Synthetic data.

in most datasets than the data dependent sanitization methods. DPM factors in previous data tuples, owing to which the metric values are better than the other mechanisms. However, DPM is computationally intensive as compared to AMHF. Moreover, DPM also has an additional preprocessing step. AMHF also incorporates randomness owing to the mechanism design, and has a performance comparable to DPM. Additionally, owing to the incorporation of the privacy parameter ϵ, the required privacy level can be tuned. Since a higher level of privacy



(a)



(b)

Figure 6.5: Resemblance Comparison for (a) Real life data (b) Synthetic data

will result in reduced utility, the tuning of the privacy level is crucial. Agents desirous of higher utility levels can set lower values of ϵ, thereby resulting in lower amount of data modification.

Dataset	Metric	NRP	AMHF	DPM
Hospital	Breach Count	0.2275 ± 0.1	0.1485 ± 0.05	0.0267 ± 0.01
	Displacement	16.0402 ± 1.83	13.4109 ± 2.15	13.8857 ± 3.06
	Resemblance	0.0798 ± 0.009	0.0826 ± 0.005	0.0786 ± 0.007
Synthetic	Breach Count	0.3197 ± 0.15	0.1681 ± 0.045	0.0337 ± 0.001
	Displacement	75.1016 ± 4.89	38.6410 ± 3.25	49.6891 ± 2.45
	Resemblance	0.0925 ± 0.005	0.0847 ± 0.003	0.0629 ± 0.002

Table 6.2: Summary of the average metric values (mean ± standard deviation).

6.5.2 Brute-Force Attack

We now take a look at a probable brute-force attack which entails exploring $(k-1)!$ permutations for $p_2, p_3, \ldots, p_k$, with additional considerations for the potential range of each p_i. Assuming p_{i-1} always comes before p_i, we restrict the number of ways of ordering of $p_2, p_3, \ldots, p_k$ to 1. Assume that each p_i is of q digits. Then we have

$$p_i \in \{\underbrace{00\ldots0}_{q-digits}, \cdots, \underbrace{99\ldots9}_{q-digits}\}.$$

Thus each p_i can be one of 10^q numbers. Then $X = p_2 p_3 \ldots p_k$ can be one of $10^{q(k-1)}$ numbers. Now as each $p_2, p_3, \cdots, p_k$ can be one of 10^q numbers, therefore, $p_1^2 + p_2^2 + \cdots + p_k^2$ value can range between 0 and $p_1^2 + (k-1)(999\cdots9)^2$.
There are $k-1$ parameters other than unique identifier and all elements are q-digits each. This implies,

$$\sum_{i=2}^{k} p_i^2 \in \{0, \cdots, (k-1) \times \underbrace{999\ldots9^2}_{q-digits}\}. \tag{6.13}$$

Since each p_i can take 10^q values, there are $10^{q(k-1)}$ possible combinations of values for p_2 to p_k for obtaining a probable value of z_2. Now the adversary knows the values of p_1, r and $g_1, g_2, \cdots, g_{n-k}$. This implies that the adversary knows z_1 and also adversary knows that z_2 can take a value formed by one of $10^{q(k-1)}$

probable combinations.

The equation for r is,

$$\begin{aligned} r^2(p_2^2 - \beta^2(z_1 + z_2)) + 2p_2(p_1^2 + z_1)r \\ +(p_1^2 + z_1)(p_1^2 + z_1 - \beta^2(z_1 + z_2)) \leq 0. \end{aligned} \tag{6.14}$$

For a fixed value of ϵ, the adversary must search over all admissible combinations of the private parameters that are consistent with the observed hash value. Consequently, the computational complexity of inversion is governed primarily by the cardinality of the private parameter space together with the collision structure induced by the proposed mapping. If we further assume that ϵ is also known, number of ways of forming the Equation 6.14 is $10^{q(k-1)}$ as only p_1 is already fixed and known.

Assuming $q = 3$ and the number of private parameters $k = 5$, the adversary is confronted with the challenge of solving 10^{12} quadratic equations out of which one equation will result in the value of hash equal to r. Higher values of q and k will require solving $10^{q(k-1)}$ equations, each representing a potential combination of $p_2, p_3, \ldots, p_k$. The iterative nature of this process highlights the computational complexity and resource demands imposed on the adversary. The above analysis assumes that the adversary performs an exhaustive (dictionary) search over the admissible private parameter space. The effective attack complexity therefore depends not only on the domain size $10^{q(k-1)}$, but also on the amount of auxiliary information available to the adversary. For example, prior knowledge of the range or values of some private parameters would reduce the search space, whereas the many-to-one nature of the proposed mapping increases ambiguity by allowing multiple candidate tuples to produce the same sanitized output. Consequently, the probability of successful recovery depends on both the available side information and the induced collision structure, with larger private domains and greater ambiguity requiring the adversary to examine substantially more candidate solutions.

6.6 Conclusion

In this chapter, we addressed the problem of inference privacy in MAS by proposing a compression based mechanism. We proposed a unified geometric formulation of utility and sanitization-induced privacy. These definitions provide a direct geometric interpretation of the privacy utility tradeoff problem. We further established that the proposed cosine similarity based measures are theoretically consistent with CRLB based metrics up to second order in the perturbation magnitude.

Further, we proposed AMHF, a decentralized sanitization function that incor-

porates the agent specific privacy requirement directly into the mechanism. The mechanism works by deriving a quadratic inequality whose solution determines the hash value. This eliminates the need for a fixed divisor and enhances unpredictability. We further proved that the quadratic equation always has a solution.

Theoretical analysis demonstrated that AMHF is non-homomorphic and that the mechanism leaks at most an ϵ-dependent amount of information about the sensitive parameters. The design intentionally promotes collisions in the hash space to strengthen inference privacy guarantees. Additionally, the computational burden imposed on a brute-force adversary grows exponentially with the number and format of private parameters.

Extensive experimentation on both synthetic and real-life hospital datasets validated the effectiveness of the proposed mechanism where AMHF achieved comparable, if not better, metric values. However, the AMHF mechanism has a significantly lower complexity, making it efficient for deployment in MAS models. Overall, this chapter establishes AMHF as a scalable and tunable compression based mechanism for enforcing inference privacy in MASs.

Chapter 7

Summary and Future Scope

This chapter provides an overview of the proposed methodologies introduced in the thesis and offers suggestions for further developing the concepts presented in the thesis.

Summary of the Results

This thesis highlights the limitations of existing methods and proposes novel inference privacy preservation mechanisms for multi-agent systems (MASs). A brief summary of the thesis, including that of the proposed mechanisms, is given below.

- **Chapter 1** introduces the multi-agent systems (MAS) and highlights the privacy risks that arise when agents share their observations with a central fusion center for distributed inference. The chapter distinguishes between data privacy and inference privacy, emphasizing the risk of unintended leakage of sensitive information through statistical inference. It formulates the inference privacy preservation problem, discusses the fundamental privacy-utility tradeoff issue, and outlines the design requirements for privacy preserving mechanisms in decentralized and resource constrained MAS environments.
- **Chapter 2** reviews the existing literature on privacy preserving mechanisms relevant to MAS. The chapter examines differential privacy (DP), dimension reduction techniques, cryptographic approaches, and federated learning-based methods, highlighting their strengths and limitations in decentralized and resource constrained MAS environments. Particular attention is given to challenges arising from streaming data, data correlations, and reconstruction attacks. The chapter also discusses various formulations of the privacy-utility tradeoff problem and analyzes the existing optimization frameworks for balancing the tradeoff. Some of the limitations identified in current approaches are listed below.

  1. The privacy guarantee of existing methods worsens upon repeated application on spatio-temporally correlated data. That is, the existing methods are not suitable for correlated data. Also, the uniform privacy budget allocation does not consider the utility of the data.
  2. Most methods require batch processing of data, that is, for the application of the mechanism, they need the whole or a major part of the dataset.
  3. The high computational cost of some of the existing methods makes them unsuitable for deployment in MAS. Consisting of low-resourced agents, MAS models cannot use computationally complex algorithms for privacy preservation.

  These shortcomings motivate the development of lightweight, decentralized, and correlation aware privacy preserving mechanisms proposed in this thesis.

- **Chapter 3** investigates the limitations of standard DP mechanisms in MAS that involve continuous location queries, such as location based services. The chapter shows that when DP with a fixed per timestep privacy budget is repeatedly applied to correlated location data, the overall privacy guarantee deteriorates due to sequential composition and spatio-temporal correlations. We theoretically prove that an ϵ-DP mechanism applied to a trace of size n' results in a cumulative privacy loss of $n'\epsilon$.

  To address this issue, we propose a perturbation generation mechanism for the enforcement of privacy for such spatio-temporally correlated data. The proposed privacy framework works in three stages. The first step involves finding neighbouring locations of the user (agent). The second step examines the predicted locations and returns those positions that can be used as a perturbed position. Finally, one of the possible locations is chosen as the perturbed location and its feasibility is tested. In case the test fails, we employ a noise mechanism to perturb the actual location. Further, the experimental results on a real life dataset prove the efficacy of the perturbation generation method for such mobile systems.

- **Chapter 4** addresses the limitations of uniform DP budget allocation in streaming MAS. It shows that distributing a fixed privacy budget uniformly across timesteps leads to inefficient noise injection and degraded utility, particularly for temporally correlated data streams. To overcome this limitation, the chapter proposes an adaptive privacy budget allocation (APBA) mechanism that dynamically allocates the privacy budget based on local signal uncertainty and the agent's influence on the global estimate. The proposed framework consists of three main components: (i) adaptive privacy budget allocation per agent, (ii) local perturbation of observations, and (iii) fusion center aggregation. Theoretical analysis establishes that APBA preserves the overall differential privacy guarantee under sequential composition while bounding estimation error and ensuring robustness to agent dropout. Experimental evaluation on real-world streaming datasets demonstrates that APBA significantly improves estimation accuracy compared to existing baselines while maintaining formal privacy guarantees, particularly in environments with high temporal variability.

- In **Chapter 5**, we focus on projection based dimensionality reduction mechanisms for inference privacy preservation. The chapter first reviews classical dimension reduction techniques like basic random projection (BRP), and neuronal random projection (NRP). BRP and NRP were initially introduced for reducing the dimension of examples in the robust concept deduction

problem. This chapter highlights the analogy between the robust concept deduction problem and that of MAS. Further, the computational and privacy limitations of BRP and NRP in streaming MAS environments is highlighted. To address these issues, a learning based dynamic projection method (DPM) is proposed, where the projection matrix is initialized once and subsequently updated using the current raw and sanitized data tuples. The method incorporates a learning rate and regularization factor to control the influence of new observations while maintaining stability and balancing privacy with utility. Theoretical analysis establishes bounds on utility preservation and shows that cumulative privacy leakage remains bounded. Experimental evaluation on both real world and synthetic datasets demonstrates that DPM achieves improved privacy-utility tradeoffs and lower computational overhead compared to existing projection based methods.

- **Chapter 6** proposes a hashing based privacy preserving mechanism for MAS, named adaptive modulo hash function (AMHF). The chapter first establishes unified definitions of utility and privacy using cosine similarity, providing a simple and interpretable measure of the privacy-utility tradeoff and showing its theoretical consistency with CRLB based metrics. Building on these definitions, the AMHF mechanism is developed to sanitize private parameters in a decentralized manner while incorporating agent specific privacy constraint. The method computes the hash value by solving a quadratic equation derived using the privacy requirement and the elements of the tuple, thereby eliminating the need for a fixed divisor and improving unpredictability. Theoretical analysis demonstrates that AMHF is non-homomorphic and leaks at most an ϵ-dependent amount of information about the sensitive parameters. Experimental evaluations on both synthetic and real world datasets show that AMHF achieves competitive privacy and utility performance compared to existing dimension reduction methods while maintaining significantly lower computational complexity, making it suitable for resource constrained multi-agent environments.

A summary comparing the four proposed mechanisms across relevant dimensions is outlined in Table 7.1.

Method	Methodology	Computational Characteristics
Perturbation method for continuous queries (Chapter 3).	Exploits mobility profile to compute feasible perturbed output. Uses Laplacian method as a fallback in the absence of candidate for output.	$\mathcal{O}(n \log n)$ for sanitization (with unsorted mobility probabilities).
Assumptions: The adversary is assumed to know the released perturbed locations and the user's mobility profile.		
Adaptive privacy budget allocation (Chapter 4).	Laplacian method using adaptively allocated budget (allocation depends on informativeness of data): provides DP over the horizon of communication.	$\mathcal{O}(n)$ for sanitization.
Assumptions: The individual components of of uncertainty and influence may be computed by the fusion center, but the fusion is center is trusted to execute normalization and influence computation. Only the sanitized components are considered released outputs. **Key Limitations:** • The differential privacy guarantees established in this chapter does not explicitly account for auxiliary information that may arise from the adaptive budget allocation process. • The mechanism is reactive and thus, a minimum reserve or horizon assumption has not been considered.		
Dynamic projection method (Chapter 5).	Dimension reduction using dynamic, data dependent matrix updation: provides reconstruction resistance.	$\mathcal{O}(nm)$ for sanitization.
Assumptions: All tuple formats are assumed to be the same for all agents. Stability, boundedness and convergence is assumed for the current choice of learning rate and regularization parameters. **Key Limitations:** • Choice of re-orthogonalization and refresh thresholds are currently heuristic.		

Method	Methodology	Computational Characteristics
Adaptive modulo hashing (Chapter 6).	Dimension reduction via hash generation using private parameters using tuple elements and privacy requirement: provides reconstruction resistance.	$\mathcal{O}(n)$ for sanitization.
Assumptions: Tuple formats are fixed with the position of private and public variables consistent across all agents. The privacy parameter is assumed to be set independently by the agent. **Key Limitations:** • Collision behaviour needs to be further analysed, with a formal analysis of the probability of collision distribution.		

Table 7.1: Brief summary of the proposed methods.

Scope for Future Work

A brief outline, describing the possible extensions of the current work to be carried out in the future with suitable model problems are presented below.

- We would like to derive tight analytical bounds characterizing the exact privacy-utility tradeoff region for the proposed methods. In this thesis, we have enforced bounds on the total privacy budget or a lower bound constraint on privacy. However, the privacy-utility tradeoff problem needs to be optimized and the exact Pareto frontier remains to be classified and visualized.

- In Chapter 4, we will further investigate privacy-preserving budget allocation mechanisms, where the statistics driving adaptation are themselves differentially private or integrated within adaptive composition frameworks such as privacy filters or privacy odometers, thereby extending the formal privacy guarantees to both the released data and the allocation procedure.

- In Chapter 5, the stability, bounded and convergence would be further analysed to implement a principled choice of learning-rate, regularization parameter, re-orthogonalization and refresh thresholds.

- In Chapter 6, we assumed that cosine similarity between the raw and the sanitized data is in the interval $[0, 1]$. This follows by the imposition that the sanitization mapping $\mathcal{M}$ preserves the sign of the inner product, i.e.,

  $$\mathbf{y}_i^T \mathcal{M}(\mathbf{y}_i) \geq 0,$$

  for all agents i. This assumption holds in Chapter 6 of our thesis by design of $\mathcal{M}$. However, the feasibility of this assumption needs to be studied for general sanitization mechanisms. As part of the future work, we would establish necessary and sufficient conditions under which this guarantee holds.

  The collision behaviour will also be further analyzed to properly understand the tradeoff between utility and privacy.

- In this thesis, we have considered an honest-but-curious (HBC) adversary. We want to further test the efficacy of our proposed algorithms against stronger adversaries, including side-information attacks, collusion among agents, adaptive reconstruction strategies and learning-based reconstruction attacks, including autoencoder and generative adversarial network (GAN) based methods.

- We have considered the temporal correlations present within the observations of an individual agent. In practical deployments, observations collected by different agents may also exhibit spatial or structural correlation. Such inter-agent dependencies can potentially be exploited by an adversary to improve the inference of one agent using the sanitized reports of neighboring agents. We will further investigate coordinated privacy mechanisms capable of jointly accounting for cross-agent dependencies.

- We will investigate the robustness of the proposed mechanisms against sophisticated sequential inference adversaries, such as hidden Markov models, particle filters, recurrent neural networks, and map-constrained Bayesian attackers. Furthermore, while the present work employs generic application-independent utility metrics, including displacement and resemblance, future studies may incorporate application-specific service-level metrics, such as route error, nearest-service accuracy, and region-level task accuracy, to provide a more operational evaluation for location-based services.

- We would also investigate nonlinear sanitization mappings like cryptographic techniques such as secure multiparty computation or homomorphic encryption (HE), as well as federated learning (FL) based mechanisms. Deploying appropriate cost reduction methods might make them strong contenders for preservation of inference privacy in the case of MAS. In case the costs can

be effectively reduced, HE and FL can be effectively deployed in MAS. This is an area of study that we would like to undertake.

- In this thesis, we assumed that the tuple size, the number of private and public parameters and their positions in the raw tuple are the same for all the agents. However, in reality, some parameters might be private for a fraction of the agents, while it may not be so for the others. This makes it necessary to handle such different tuple sizes and formats in the case of a MAS. Also, we have assumed numerical raw data in the thesis. In future, we would incorporate heterogeneous data types as well as agent-specific requirements like different tuple structures, size, and location of private parameters.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.

[3] Abdullah Albelaihy and Jonathan Cazalas. Privacy preserving queries for lbs: Hash function secured (hfs). In *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)*, pages 7–12. IEEE, 2017.

[4] Tristan Allard, Hira Asghar, Gildas Avoine, Christophe Bobineau, Pierre Cauchois, Elisa Fromont, Anna Monreale, Francesca Naretto, Roberto Pellungrini, Francesca Pratesi, et al. Analyzing and explaining privacy risks on time series data: ongoing work and challenges. *ACM SIGKDD Explorations Newsletter*, 26(1):49–58, 2024.

[5] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914, 2013.

[6] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in neural information processing systems*, 34:17455–17466, 2021.

[7] Anubhuti and Harjot Kaur. Role of multi-agent systems in health care: A review. *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 2*, pages 367–378, 2022.

[8] Peter Bodik, Wei Hong, Carlos Guestrin, Sam Madden, Mark Paskin, and Romain Thibaux. Intel Lab Data. `https://db.csail.mit.edu/labdata/labdata.html`, 2004. MIT CSAIL.

[9] Sandeep Chandra Bollepalli, Ashish Kumar Sahani, Naved Aslam, Bishav Mohan, Kanchan Kulkarni, Abhishek Goyal, Bhupinder Singh, Gurbhej Singh, Ankit Mittal, Rohit Tandon, et al. An optimized machine learning model accurately predicts in-hospital outcomes at admission to a cardiac unit. *Diagnostics*, 12(2):241, 2022.

[10] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth.

Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[11] Erik Buchholz, Alsharif Abuadbba, Shuo Wang, Surya Nepal, and Salil Subhash Kanhere. Reconstruction attack on differential private trajectory protection mechanisms. In *Proceedings of the 38th annual computer security applications conference*, pages 279–292, 2022.

[12] Bodhi Chakraborty, Shekhar Verma, and Krishna Pratap Singh. Temporal differential privacy in wireless sensor networks. *Journal of Network and Computer Applications*, 155:102548, 2020.

[13] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. A predictive differentially-private mechanism for mobility traces. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 21–41. Springer, 2014.

[14] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[15] Rinku Dewri. Local differential perturbations: Location privacy under approximate knowledge attackers. *IEEE Transactions on Mobile Computing*, 12(12):2360–2372, 2012.

[16] Angela Di Fazio. Enhancing privacy in recommender systems through differential privacy techniques. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 1348–1352, 2024.

[17] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[18] Roel Dobbe, Ye Pu, Jingge Zhu, Kannan Ramchandran, and Claire Tomlin. Customized local differential privacy for multi-agent distributed optimization. *arXiv preprint arXiv:1806.06035*, 2018.

[19] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

[20] Xiaoming Duan, Zhe Xu, Rui Yan, and Ufuk Topcu. Privacy-utility tradeoffs against limited adversaries. *IEEE Transactions on Automatic Control*, 69(1):519–526, 2023.

[21] John C Duchi, Michael I Jordan, and Martin J Wainwright. Privacy aware learning. *Journal of the ACM (JACM)*, 61(6):1–57, 2014.

[22] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

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

[24] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

[25] Berk Erisen. Wind Turbine SCADA Dataset. Kaggle, 2018. Available on Kaggle.

[26] Xiaoyu Fan, Guosai Wang, Kun Chen, Xu He, and Wei Xu. Ppca: Privacy-preserving principal component analysis using secure multiparty computation (mpc). *arXiv preprint arXiv:2105.07612*, 2021.

[27] Natasha Fernandes, Annabelle McIver, and Carroll Morgan. The laplace mechanism has optimal utility for differential privacy over continuous queries. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2021.

[28] William Ford. Chapter 17 - implementing the qr decomposition. In William Ford, editor, *Numerical Linear Algebra with Applications*, pages 351–378. Academic Press, Boston, 2015.

[29] Agostino Forestiero. Multi-agent recommendation system in internet of things. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 772–775. IEEE, 2017.

[30] Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.

[31] David Froelicher, Juan R Troncoso-Pastoriza, Apostolos Pyrgelis, Sinem Sav, Joao Sa Sousa, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. Scalable privacy-preserving distributed learning. *arXiv preprint arXiv:2005.09532*, 2020.

[32] Jie Fu, Zhili Chen, and Xiao Han. Adap dp-fl: Differentially private federated learning with adaptive noise. In *2022 IEEE international conference on trust, security and privacy in computing and communications (TrustCom)*, pages 656–663. IEEE, 2022.

[33] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947, 2020.

[34] Puspanjali Ghoshal and Ashok Singh Sairam. Dimension reduction via random projection for privacy in multi-agent systems. *arXiv preprint arXiv:2412.04031*, 2024.

[35] Atefeh Gilani, Juan Felipe Gomez, Shahab Asoodeh, Flavio Calmon, Oliver Kosut, and Lalitha Sankar. Optimizing noise distributions for differential privacy. In *Forty-second International Conference on Machine Learning*, 2025.

[36] Vehbi C Gungor and Gerhard P Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on industrial electronics*, 56(10):4258–4265, 2009.

[37] Muhammad Zia Ul Haq, Muhammad Zahid Khan, Haseeb Ur Rehman, Gulzar Mehmood, Ahmed Binmahfoudh, Moez Krichen, and Roobaea Alroobaea. An adaptive topology management scheme to maintain network connectivity in wireless sensor networks. *Sensors*, 22(8):2855, 2022.

[38] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams. Foundations for smarter cities. *IBM Journal of Research and Development*, 54(4):1–16, 2010.

[39] István Hegedus and Márk Jelasity. Distributed differentially private stochastic gradient descent: An empirical study. In *2016 24th Euromicro international conference on parallel, distributed, and network-based processing (PDP)*, pages 566–573. IEEE, 2016.

[40] A Hero and Jeffrey A Fessler. A recursive algorithm for computing cramer-rao-type bounds on estimator covariance. *IEEE Transactions on Information Theory*, 40(4):1205–1210, 1994.

[41] Junyuan Hong, Zhangyang Wang, and Jiayu Zhou. Dynamic privacy budget allocation improves data efficiency of differentially private gradient descent. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 11–35, 2022.

[42] Chenxi Huang, Chaoyang Jiang, and Zhenghua Chen. Shuffled differentially private federated learning for time series data analytics. In *2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*, pages 1023–1028. IEEE, 2023.

[43] Sharu Theresa Jose and Osvaldo Simeone. An information-theoretic analysis of the cost of decentralization for learning and inference under privacy constraints. *Entropy*, 24(4):485, 2022.

[44] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12):1719–1733, 2007.

[45] Sanjay Kariyappa, Ousmane Amadou Dia, and Moinuddin K. Qureshi. Enabling inference privacy with adaptive noise injection. *ArXiv*, abs/2104.02261, 2021.

[46] Shahrzad Kiani, Nupur Kulkarni, Adam Dziedzic, Stark Draper, and Franziska Boenisch. Differentially private federated learning with time-adaptive privacy spending. *arXiv preprint arXiv:2502.18706*, 2025.

[47] Bogdan Kulynych, Juan F Gomez, Georgios Kaissis, Flavio du Pin Calmon, and Carmela Troncoso. Attack-aware noise calibration for differential privacy. *Advances in Neural Information Processing Systems*, 37:134868–134901, 2024.

[48] Yeon-Ji Lee, Na-Yeon Shin, and Il-Gu Lee. Privacy-preserving data sharing via pca-based dimensionality reduction in non-iid environments. *Electronics*, 14(13):2711, 2025.

[49] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. volume 60, pages 1–35. ACM New York, NY, USA, 2013.

[50] Saber Malekmohammadi, Yaoliang Yu, and Yang Cao. Noise-aware algorithm for heterogeneous differentially private federated learning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34461–34498, 2024.

[51] Chiara Marcolla, Victor Sucasas, Marc Manzano, Riccardo Bassoli, Frank H. P. Fitzek, and Najwa Aaraj. Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10):1572–1609, 2022.

[52] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. Pmlr, 2017.

[53] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[54] Matei Moldoveanu and Abdellatif Zaidi. In-network learning: Distributed training and inference in networks. *Entropy*, 25(6), 2023.

[55] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*, pages 709–720. Springer, 2010.

[56] Ke Pan and Kaiyuan Feng. Differential privacy-enabled multi-party learning with dynamic privacy budget allocating strategy. *Electronics*, 12(3):658, 2023.

[57] Marios Papachristou and M Amin Rahimian. Differentially private distributed estimation and learning. *IISE Transactions*, 57(7):756–772, 2025.

[58] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends$\hat{W}$ in Theoretical Computer Science*, 10(4):283–424, 2016.

[59] M. Pipattanasomporn, H. Feroze, and S. Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *2009 IEEE/PES Power Systems Conference and Exposition*, pages 1–8, 2009.

[60] Bernardo Pulido-Gaytan, Andrei Tchernykh, Jorge M Cortés-Mendoza, Mikhail Babenko, Gleb Radchenko, Arutyun Avetisyan, and Alexander Yu Drozdov. Privacy-preserving neural networks with homomorphic encryption: C hallenges and opportunities. *Peer-to-Peer Networking and Applications*, 14(3):1666–1691, 2021.

[61] Aradhana Sahu and Samarendra Mohan Ghosh. Review paper on secure hash algorithm with its variants. *International Journal of Technical Innovation in Modern Engineering & Science*, 3(05), 2017.

[62] Deblina Mazumder Setu, Tania Islam, Md Erfan, Samrat Kumar Dey, Md Rashid Al Asif, and Md Samsuddoha. A comprehensive strategy for identifying plagiarism in academic submissions. *Journal of Umm Al-Qura University for Engineering and Architecture*, 16(2):310–325, 2025.

[63] William Stallings. *Cryptography and network security, 4/E*. Pearson Education India, 2006.

[64] Meng Sun and Wee Peng Tay. Inference and data privacy in iot networks. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2017.

[65] Meng Sun and Wee Peng Tay. On the relationship between inference and data privacy in decentralized iot networks. *IEEE Transactions on Information Forensics and Security*, 15:852–866, 2019.

[66] Nazanin Takbiri, Amir Houmansadr, Dennis L Goeckel, and Hossein Pishro-Nik. Matching anonymized and obfuscated time series to users' profiles. *IEEE Transactions on Information Theory*, 65(2):724–741, 2018.

[67] Santosh S Vempala. *The Random Projection Method*, volume 65. American Mathematical Soc., 2005.

[68] Chong Xiao Wang, Yang Song, and Wee Peng Tay. Preserving parameter privacy in sensor networks. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1316–1320. IEEE, 2018.

[69] Chong Xiao Wang, Yang Song, and Wee Peng Tay. Arbitrarily strong utility-privacy tradeoff in multi-agent systems. *IEEE Transactions on Information Forensics and Security*, 16:671–684, 2020.

[70] Chong Xiao Wang and Wee Peng Tay. Data-driven regularized inference privacy. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[71] Chong Xiao Wang, Wee Peng Tay, and Yang Song. Maximum privacy under perfect utility in sensor networks. In *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE, 2020.

[72] Jie Wang, Zhiju Zhang, Jing Tian, and Hongtao Li. Local differential privacy federated learning based on heterogeneous data multi-privacy mechanism. *Computer Networks*, 254:110822, 2024.

[73] Jun Wang, Rongbo Zhu, Shubo Liu, and Zhaohui Cai. Node location privacy protection based on differentially private grids in industrial wireless sensor networks. *Sensors*, 18(2):410, 2018.

[74] Zhiqiang Wang, Xinyue Yu, Qianli Huang, and Yongguang Gong. An adaptive differential privacy method based on federated learning. *arXiv preprint arXiv:2408.08909*, 2024.

[75] Yonghui Xiao and Li Xiong. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1298–1309, 2015.

[76] Xudong Yang, Ling Gao, Jie Zheng, and Wei Wei. Location privacy preservation mechanism for location-based service with incomplete location data. *IEEE Access*, 8:95843–95854, 2020.

[77] Jiaojiao Zhang, Dominik Fay, and Mikael Johansson. Dynamic privacy allocation for locally differentially private federated learning with composite objectives. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9461–9465. IEEE, 2024.

[78] Mingwu Zhang, Shuo Huang, Gang Shen, and Yuntao Wang. Ppnnp: A privacy-preserving neural network prediction with separated data providers using multi-client inner-product encryption. *Computer Standards & Interfaces*, 84:103678, 2023.

[79] Tao Zhang, Tianqing Zhu, Renping Liu, and Wanlei Zhou. Correlated data in differential privacy: definition and analysis. *Concurrency and Computation: Practice and Experience*, 34(16):e6015, 2022.

[80] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning, ICML 2022*, 2022.

[81] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[82] Shuheng Zhou, John Lafferty, and Larry Wasserman. Compressed and privacy-sensitive sparse regression. *IEEE Transactions on Information Theory*, 55(2):846–866, 2009.

[83] Shuheng Zhou, Katrina Ligett, and Larry Wasserman. Differential privacy with compression. In *2009 IEEE International Symposium on Information Theory*, pages 2718–2722. IEEE, 2009.

[84] Bian Zhu and Ling Niu. A privacy-preserving federated learning scheme with homomorphic encryption and edge computing. *Alexandria Engineering Journal*, 118:11–20, 2025.

[85] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

Publications

Publications from this thesis

Journals

1. **Puspanjali Ghoshal, Mohit Dhaka, Ashok Singh Sairam** "On the effectiveness of differential privacy to continuous queries". Service Oriented Computing and Applications (SOCA), Springer.

2. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Dimension Reduction via Random Projection for Privacy in Multi-Agent Systems" (Under Review).

3. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Preserving Inference Privacy in Multi-Agent Systems with Adaptive Modulo Hash Function" (Under Review).

Conferences

1. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Utility Aware Adaptive Privacy Budget Allocation for Streaming Multi-Agent Systems". In Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems 2026 (AAMAS '26). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.

2. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Dynamic Projection Method: A Learning Based Approach For Preserving Inference Privacy", In Proceedings of IEEE Global Communications Conference 2025 (GLOBECOM '25), Taipei, Taiwan, 2025.

3. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Relationship between Noise addition and Compression based methods for Differential Privacy in Multi-Agent Systems" at IEEE ANTS 2023 at MNIT Jaipur (17-20 December 2023).

Publications outside this thesis

Published

1. **Puspanjali Ghoshal, Charan Annadurai, Axel Sikora, Ashok Singh Sairam**, "Privacy Preservation with Decentralized Authentication in Multi-Agent Systems" at IEEE ANTS 2024 at IIT Guwahati (15-18 December 2024).

Under Preparation

1. **Puspanjali Ghoshal, Ashok Singh Sairam**, "Adaptive and Attack-Aware Federated Learning with Dynamic Privacy Control" (Manuscript Under Preparation).
2. **Puspanjali Ghoshal, Dominik Welte, Axel Sikora, Ashok Singh Sairam**, "Distributed Authentication and Privacy Scheme" (Manuscript Under Preparation).
3. **Puspanjali Ghoshal**, "Lightweight Inference-Privacy Preserving Distributed Estimation via Task-Specific Minimal Statistics" (Under Review).