Northumbria University
NEWCASTLE

PRESERVING INDIVIDUAL PRIVACY IN UBIQUITOUS E-COMMERCE ENVIRONMENTS: A UTILITY PRESERVING APPROACH FOR USER-BASED PRIVACY CONTROL

J NEERA

PhD

2022

PRESERVING INDIVIDUAL PRIVACY IN UBIQUITOUS E-COMMERCE ENVIRONMENTS: A UTILITY PRESERVING APPROACH FOR USER-BASED PRIVACY CONTROL

JEYAMOHAN NEERA

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of Doctor of Philosophy

Faculty of Environment and Engineering

October 2022

Abstract

Applications such as e-commerce, smart home appliances, and healthcare systems, amongst other things, have become part and parcel of our daily lives. The data aggregated through these applications combined with state-of-the-art machine learning approaches have even increased the widespread uptake of these applications. However, such data aggregation and analytical practices have raised privacy concerns among users. Privacy-preserving machine learning models mitigate these concerns through private data aggregation and analytical techniques.

The first objective of this thesis is to design a privacy-preserving data aggregation and analytical approach for recommendation systems. Recommendation systems rely heavily on behavioural and preferential data of a user to produce accurate recommendations. Aggregation of such data can reveal sensitive information about users to the Third-Party Service Providers (TPSPs). We start with designing a recommendation system that uses Local Differential Privacy (LDP) based input data perturbation mechanism to perturb users' ratings locally before sending it to the TPSP. Hence, the TPSP aggregates only the perturbed ratings and has no access to original ratings. This approach protects a user's privacy from TPSPs who aggregate ratings to infer any sensitive information. Next, we propose an LDP-based hybrid recommendation framework to protect users' privacy from TPSPs who aggregate both ratings and reviews. We propose to perturb user ratings and pre-process user reviews at the user-side before sending them to the TPSP. Such an approach ensures that the TPSP cannot aggregate the original ratings or reviews from the users. However, these approaches still do not protect a user's privacy from TPSPs who collect implicit feedback to predict a user's preferences. Hence, we design an LDP-based federated matrix factorization for implicit feedback. We motivate the idea of stochastic gradient perturbation using the Bounded Laplace (BLP) mechanism to ensure strong privacy protection for users. The second objective of this thesis is to design a privacy-preserving untraceable TPSP-based payment protocol. A TPSP-based payment system does not protect a customer's privacy in the face of an untrustworthy TPSP. Customers cannot make transactions anonymously as the TPSP collects detailed transaction-related information. TPSP uses this information to create a comprehensive behaviour profile of each customer, based on which TPSP can deduce sensitive information about a customer's lifestyle. Hence we propose an untraceable payment system in this thesis to tackle this problem.

Contents

Abstract i

Acronyms vii

Acknowledgements viii

Declaration ix

1 Introduction 1

1.1 Background and Motivation 2

1.2 Aims 4

1.3 Contributions 5

1.4 Dissemination 6

1.5 Thesis Outline 7

2 Background 8

2.1 E-Commerce Systems 8

2.1.1 Introduction to Recommendation Systems 9

2.1.2 Importance of Privacy in E-commerce systems 16

2.2 Privacy Protection Technologies 19

2.2.1 Cryptographic Primitives 19

2.2.2 Differential Privacy 25

2.2.3 Federated Learning 28

2.3 Existing Work 31

2.3.1 Untraceable Payment Systems 31

2.3.2 Privacy Protection in Recommendation Systems 33

2.3.3 Federated Learning-based Recommendation Systems 36

3 LDP-based Collaborative Filtering Recommendation System 38

3.1 Local Differential Privacy Recommendation with BLP and MoG 40

3.1.1 LDP Rating Perturbation 41

3.1.2 Noise Estimation with Mixture of Gaussians 48

3.1.3 Expectation Maximisation for MoG . . . 50
3.2 Evaluation . . . 52
3.2.1 Datasets . . . 52
3.2.2 Evaluation Metrics . . . 53
3.2.3 Results . . . 54
3.3 Conclusion . . . 60

4 LDP-based Hybrid Recommendation Model **61**
4.1 Hybrid Recommendation System with LDP . . . 62
4.1.1 Input Rating Perturbation at User-side . . . 62
4.1.2 Review Pre-processing at User-side . . . 63
4.1.3 Review Tokenizing using Bidirectional Encoder Representations from Transformers . . . 63
4.1.4 Perturbed Rating Classification at Server-side . . . 64
4.1.5 Multi Class Classification Using CNN . . . 64
4.1.6 Matrix Factorization with Mixture of Gaussian Model . . . 66
4.1.7 Predicted Ratings Combination . . . 67
4.2 Evaluation . . . 67
4.2.1 Datasets . . . 67
4.2.2 Evaluation Metrics . . . 68
4.2.3 Results . . . 68
4.3 Conclusion . . . 71

5 Federated Matrix Factorization with Local Differential Privacy **72**
5.1 Matrix Factorization for Implicit Feedback . . . 74
5.2 Federated Matrix Factorization for Implicit Feedback . . . 76
5.3 LDP-based Federated Learning Matrix Factorization for Implicit Feedback . . . 77
5.4 Evaluation . . . 82
5.4.1 Datasets . . . 82
5.4.2 Evaluation Metric . . . 82
5.4.3 Results . . . 83
5.5 Discussion . . . 86

5.6 Conclusion . . . 87

6 Untraceable Payment System **89**

6.1 Third-Party based Payment Systems . . . 90

6.2 An Untraceable Third-party based Payment System . . . 92

6.2.1 Overview of the System . . . 92

6.2.2 Registration . . . 93

6.2.3 ZKP Mutual Authentication . . . 95

6.2.4 Purchase Request . . . 97

6.2.5 Token Withdrawal . . . 98

6.2.6 Token Issuance . . . 99

6.2.7 Payment By Token . . . 100

6.2.8 Token Deposit . . . 100

6.2.9 Refunding . . . 101

6.3 Formal Proof for Security and Privacy Analysis . . . 102

6.3.1 Untraceability . . . 102

6.3.2 Unforgeability . . . 104

6.3.3 Exculpability . . . 105

6.4 Experimental Evaluation for Security and Privacy Analysis . . . 106

6.4.1 Protocol Modelling in ProVerif Tool . . . 107

6.4.2 Formalising the Protocol . . . 108

6.4.3 Protocol Evaluation . . . 112

6.4.4 Computational Complexity Analysis . . . 114

6.5 Conclusion . . . 116

7 Conclusion **117**

7.1 Overview of Contributions . . . 117

7.2 Limitations . . . 119

7.3 Future Works . . . 120

Acronyms **122**

References **123**

List of Figures

1 Decomposition of Rating Matrix into User/Item Latent Factor Matrices 11

2 Hybrid Recommendation System with CF Algorithm and Sentiment Classification Model 16

3 A Simplified Blind Signature Scheme 20

4 Federated Learning Model with a Centralised Server 29

5 LDP-based MF Recommendation with MoG 41

6 Laplace vs Bounded Laplace Noise Distribution 55

7 Bounded Laplace Mechanism vs Laplace Mechanism RMSE Comparison 56

8 MoG vs SVD Prediction Model RMSE Comparison 57

9 PG-MF vs BLP-MoG-MF RMSE Comparison for Movielens 57

10 PG-MF vs BLP-MoG-MF F1-Score Comparison for Movielens 58

11 BLP-MoG-MF vs ISGD RMSE Comparison 59

12 BLP-MoG-MF vs ISGD RMSE Comparison 59

13 Proposed LDP-based Hybrid Recommendation System Training Architecture . . 63

14 Hybrid Sentiment Analysis 65

15 BERT-MF-MoG vs MF-MoG RMSE Comparison 69

16 BERT-MF-MoG vs MF-MoG F-Score Comparison 69

17 β-RMSE Comparison 70

18 β-F-score Comparison 70

19 LDP-based Federated Matrix Factorization for Implicit Feedback 78

20 HR@10 When Varying Privacy Budget for Small, Medium and Large Movielens Dataset 84

21 HR@10 When Varying Privacy Budget for Small, Medium and Large Jester Dataset 85

22 HR@10 When Varying Number of Iterations for Movielens and Jester Dataset. Privacy Budget $\varepsilon = 3$ and User/Item Set Size is Medium. 86

23 HR@10 Performance for Movielens and Jester Datasets of Varying User/Item Set Size. Privacy Budget $\varepsilon = 3$ and Number of Iterations $k = 300$. 87

24 Workflow of the Generic TPSP-based Mobile Payment System 91
25 Workflow of the Proposed Untraceable Payment System 93
26 Public Identity Keys Validation using Blind Signature 95
27 Mutual Authentication Scheme 98
28 Overview of the Proposed Untraceable Payment System 101
29 Computational Cost Comparison 115

List of Tables

1 Example Rating Matrix 10

2 LDP and MF Notations 40
3 Datasets for BLP-MoG-MF Evaluation 52
4 Confusion Metric. 53
5 Comparison of Communication Cost for Movielens 60

6 Perturbed Rating Classification 65
7 Datasets for BERT-MF-MoG Evaluation 67

8 Datasets for Federated Learning Evaluation 82
9 User/Item Set Size for Movielens and Jester Datasets 82

10 Pi-Calculus Grammar Notations 108
11 Analysis of Secrecy Property 112
12 Analysis of Authenticity Property 114
13 Computational Performance Analysis 115

Acronyms

ALS	Alternating Least Squares
BERT	Bidirectional Encoder Representations from Transformers
CB	Content-based Filtering
CF	Collaborative Filtering
CNN	Convolutional Neural Networks
DP	Differential Privacy
EM	Expectation Maximization
HS	Hybrid Systems
ISGD	Input Perturbation Method
KNN	K-Nearest Neighbour
LDP	Local Differential Privacy
MF	Matrix Factorization
PG-MF	Private Gradient-Matrix Factorization
RMSE	Root Mean Squared Error
SGD	Stochastic Gradient Descent
TPSP	Third-Party Service Provider
ZKP	Zero-Knowledge Proof

Acknowledgements

This PhD has been an extraordinary journey for me personally. It would not have been possible for me to complete this thesis without the support of exceptional individuals and institutions. I am grateful to everyone who has given me the confidence, resources, funding, aid, support and courage to embark on this research.

First and foremost, I would like to express my sincere gratitude to my supervisor Dr Xiaomin Chen for introducing me to the area of privacy-preserving in machine learning and for her continuous support during my PhD journey and for being extremely patient with me. She offered me valuable feedback and guided me kindly so that I could explore this topic I am passionate about.

My sincere gratitude also goes to Prof. Nauman Aslam, for his support and guidance, especially during critical periods of my journey. His guidance and advice have helped me in all aspects of my research and writing of this thesis.

I am also grateful to my collaborators, the other colleagues from the Department of Computer Information Sciences of Northumbria University and fellow PhD students for the support and encouraging words they have given me throughout.

Finally, I could not have gone far without the love, support and encouragement of my friends and family. They have been the supporting pillars of my life, and their support has been consistently warm and unconditional. I would not have been able to be where I am right now without them.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the University Ethics Committee on 31st of January 2019 (Ref No: 14123).

I declare that the Word Count of this thesis is 38202.

Name: J Neera

Date: 30 October 2022

Chapter 1

Introduction

Data has become a valuable commodity that offers significant utility when combined with machine and deep learning models. As a result, we have various applications such as recommendation systems, health monitoring systems, payment systems, autonomous vehicles etc., which use such models to help us improve the quality of our lives. However, the data aggregated and used in these models are often sensitive and can cause severe privacy risks. Some data aggregators might even use insights from these data to manipulate the decisions and choices of users (Cadwalladr and Graham-Harrison, 2018). Hence, users are increasingly becoming curious about taking ownership of their data and inherently wanting to monetise it while limiting access (Angwin et al., 2016).

Even though recent case studies have enhanced the privacy concerns related to unethical data aggregation and analytical practices, such practices are beneficial in various contexts. For example, recommendation systems rely on users' purchases and preferential data to produce recommendations of items that might interest them. Smart home applications collect a user's behaviour-related data to optimise specific devices' usage and avoid over power consumption. In addition, data collected over a large set of users can be valuable in revealing interesting new information for research related to various social, economic or medical issues.

Many governments have introduced legal restrictions to regulate the collection, distribution and usage of sensitive data relating to a user. For example, HIPAA (Health Insurance Portability and Accountability Act) outlines the lawful use and disclosure of protected health information in the

United States. Similarly, the General Data Protection Regulation (GDPR) law in the UK and the EU requires companies to comply with data protection and privacy requirements in the UK and the European Economic Area. Even though such legal means restrict unethical data aggregation and analytical practices of untrustworthy Third-Party Service Provider (TPSP), they also impact the ability of the machine and deep learning models to benefit from aggregated data.

As a result, organisations face a range of ethical and legal dichotomies while being unable to let go of such data aggregation and analytical practices. Hence, the privacy-utility trade-off has become an important issue to be tackled in many domains. The frameworks developed in this thesis focus on privacy-utility trade-offs in e-commerce environments, concentrating mainly on private data aggregation and analytical practices of recommendation and TPSP-based payment systems.

1.1 Background and Motivation

Privacy Preserving Recommendation Systems

Recommendation systems are among the most popular artificial intelligence applications, essential for an e-commerce platform. For example, users receive personalised recommendations from e-commerce platforms like Amazon or eBay. Generally, a massive amount of data is aggregated from users, allowing the recommendation system to model user preferences on non-observed items precisely. Collaborative Filtering (CF) algorithms such as Matrix Factorization (MF) are predominantly used in recommendation systems to produce these personalised recommendations.

These large-scale data aggregation and analytical practices of e-commerce platforms have raised privacy concerns from users, as most leading e-commerce platforms sell products ranging from movies and books to adult toys and health gadgets. These platforms collect user data such as location history, demographic information, historical purchases, search history, clicks, views, ratings, reviews, etc., to precisely portray a user's preferences. However, these data also can be used to infer sensitive information regarding a user. For example, a retail company was able to predict the pregnancy of their female customers through analysis of a customer's purchase history (Duhigg, 2012). Some researchers have also demonstrated how analysing an individual's historical ratings can reveal sensitive information such as a user's political preference, medical conditions and even

religious dispositions (Narayanan and Shmatikov, 2008).

Hence, more and more users are unwilling to disclose their personal information to the TPSP. The risk of intentional or accidental privacy leakage through trustworthy and untrustworthy TPSPs has called for private data aggregation and analytical practices. Differential Privacy (DP) is a popular tool that can guarantee privacy protection even when the adversary owns a considerable amount of auxiliary information about users (Dwork, 2008). Most of the existing works on DP-based privacy protection methods focus on protecting the privacy of users against a third-party adversary and assume that the risk of a TPSP causing privacy violation is minimal. Unfortunately, many TPSPs are apt to gather more data from users than they need and continue to procure sensitive information about users' behaviour for their own added benefits.

Therefore, Local Differential Privacy (LDP) (Duchi et al., 2013) has attracted much attention as it can provide a strong privacy guarantee in a setting where TPSPs are untrustworthy. Many researchers (Berlioz et al., 2015; Shin et al., 2018; Hua et al., 2015) have adopted LDP to protect the privacy of users in recommendation systems. Each user adds noise to their data locally in LDP-based privacy protection models and forwards the perturbed data to the TPSP. As the original data never leaves the user devices, users are guaranteed plausible deniability (Dwork, 2008). Nevertheless, users cannot deny the fact that some information was sent to the TPSP. Additionally, the recommendation accuracy is lower compared to DP-based recommendation systems, as DP-based recommendation systems perturb a query output, whereas LDP-based recommendation systems add noise to each data point. Hence, adopting LDP in recommendation systems causes low data utility for the TPSP.

Untraceable Payment Systems

TPSP-based payment systems have become immensely popular among users as they can conveniently make in-store and online payments using their smartphones. Generally, a TPSP who acts as an intermediary between customers and merchants owns and operates these payment systems. They provide a unified platform to carry out a fair exchange of goods and services for customers and merchants. However, these TPSPs can link multiple transactions to one customer using the aggregated transaction information and build a comprehensive customer purchase profile. Such profiles are valuable for marketing and advertising purposes as they aid TPSPs in understanding

customers' purchasing behaviour and expectations. Even though customers and merchants could benefit from insights derived from these data, there is no denying that ensuring customer transactions are untraceable by these TPSPs is becoming an essential privacy and security requirement for TPSP-owned payment systems.

Several works have proposed a variety of centralised payment systems where they use cryptographic primitives such as blind signature (Chaum, 1983; Fan and Lei, 2002; Baseri et al., 2013), zero-knowledge proof (Erway et al., 2010), certificate-less signature schemes (Zhang et al., 2011) and group signatures (Lysyanskaya and Ramzan, 1998) to ensure untraceability of transactions. However, these works assume that the TPSP is trustworthy and will not misuse customer data. Additionally, they use the services of third-party anonymous identity providers to hide the real identity of the customers, which imposes more complexity on the existing system architecture. Such an approach also allows risks of further privacy and security violations. Furthermore, the merchants involved in these works are considered honest participants, but they can perform malicious attacks such as double-spending. In addition, most of these works use more complex and computationally intensive cryptographic primitives that are not suitable for face-to-face payment scenarios where customers use mobile devices to carry out transactions.

1.2 Aims

The **Aim** of our thesis is:

- To design and develop an effective and efficient privacy-preserving recommendation and TPSP-based payment frameworks for e-commerce platforms.

We achieve this aim by decomposing it into four objectives:

- To design an effective input perturbation mechanism for CF-based recommendation systems that protects the user's privacy from the TPSP while offering optimal utility to the TPSP.
- To design a privacy-preserving hybrid CF-based recommendation that uses ratings and reviews as input while offering optimal utility to the TPSP.
- To design a privacy-preserving federated MF-based recommendation system which can hide the user-item interaction data from the TPSP.

- To design a secure and privacy-preserving TPSP-based payment system so that the TPSP is not be able to track any transaction-related information.

1.3 Contributions

The work proposed in this thesis will contribute significantly to the current body of research, enabling a privacy-based approach to data aggregation and analytical practices in the e-commerce domain by introducing privacy-preserving recommendation and payment frameworks. We highlight the contributions as follows. Contributions $1-3$ propose various approaches for a privacy-preserving recommendation framework, while contribution 4 proposes a privacy-preserving TPSP-based payment system.

1. **LDP-based Collaborative Filtering Recommendation System**: First, we introduce BLP as the input rating perturbation mechanism and MF with MoG as a noise estimation component to increase the utility of the recommendation system while offering strong privacy protection to users from untrustworthy TPSPs. We provide a sufficient condition for BLP to satisfy ε-local differential privacy in MF-based recommendation systems. Since the probability density function of BLP noise is conditional on the input rating matrix, we also derive a closed-form probability density function for noise drawn from BLP for a given dataset.

2. **LDP-based Hybrid Recommendation Model**: We propose a privacy-focused recommendation framework for hybrid recommendation models that use user ratings and reviews. We perturb the user's original ratings locally using BLP before sending them to the TPSP. We also ensure that the TPSP cannot infer sensitive information from the user reviews by tokenising the reviews locally. Additionally, the perturbed ratings are used as the input sentiment labels, preventing the TPSP from learning a user's actual sentiment corresponding to the tokenized review. Hence, our proposed framework preserves users' privacy even when a TPSP aggregates ratings and reviews and, at the same time, further improves the utility of the recommendation system.

3. **Federated Matrix Factorization with Local Differential Privacy**: We introduce BLP as an iterative input perturbation mechanism to perturb the stochastic gradient of a federated matrix factorization algorithm which uses implicit feedback as input. We perturb the

stochastic gradient locally on the user-side before sending them to the TPSP. We provide a sufficient condition for BLP to satisfy ε-local differential privacy when perturbing the stochastic gradient. We also empirically evaluate the role of privacy budget ε, the number of iterations k and the size of the user/item set plays in enhancing the predictive accuracy of the proposed recommendation system.

4. **Untraceable Payment System**: We propose an untraceable TPSP-based payment system to address the shortcomings of existing centralised untraceable payment systems. Our proposed model does not reveal any transaction-related information between a customer and a merchant to the TPSP. We still maintain the role of the TPSP in our proposed protocol as it is essential to ensure that malicious adversaries are not undermining any legal controls or carrying out illegal transactions. The proposed payment system satisfies security and privacy requirements such as untraceability, double-spending detection, exculpability, confidentiality, authenticity and unforgeability. We formally analyse our untraceable payment system using the automated verification tool Proverif. We also ensure that our proposed untraceable payment system incurs a little computational cost on the customer side.

1.4 Dissemination

To date, the work in this thesis have been disseminated via the following conferences and journal publications:

1. Neera, J., Chen, X. and Aslam, N., (2019), Local Differentially Private Matrix Factorization For Recommendations. In 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), (pp. 1-5). doi:https://doi.org/10.1109/SKIMA47702.2019.8982536.

2. Neera, J., Chen, X., Aslam, N. and Shu, Z., (2020), June. Local Differentially Private Matrix Factorization with MoG for Recommendations. In IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), (pp. 208-220). doi:https://doi.org/10.1007/978-3-030-49669-2_12.

3. Neera, J., Chen, X., Aslam, N., Wang, K. and Shu, Z., (2021), Private and Utility Enhanced Recommendations with Local Differential Privacy and Gaussian Mixture Model. IEEE

Transactions on Knowledge and Data Engineering. doi:https://doi.org/10.1109/TKDE.2021.3126577

4. Neera, J., Chen, X., Aslam, N., Isaac, B. and Obrien, E., (2022), A Local Differential Privacy based Hybrid Recommendation Model with BERT and Matrix Factorization. In 2022 International Conference on Security and Cryptography (SECRYPT).

5. Neera, J., Chen, X. and Aslam, N., (2022), An Untraceable Mobile Payment System with Partially Blind Signature and Zero-Knowledge Proof. IEEE Transactions on Dependable and Secure Computing (Under review).

1.5 Thesis Outline

The structure of the thesis is as follows:

- Chapter 2 summarises the background and motivation of this thesis. We start with setting out the foundation of this thesis via preliminaries. Then we review the existing works and discuss their strengths and limitations.
- Chapter 3 explains the BLP mechanism used to perturb user ratings in a MF-based recommendation system and the noise estimation component included at the TPSP's end to further enhance recommendation accuracy.
- In Chapter 4, we further evaluate the application of BLP as an input perturbation mechanism in a hybrid recommendation system. We present a hybrid recommendation system architecture that integrates a sentiment analysis model with a CF algorithm.
- Chapter 5 explores the application of BLP as a stochastic gradient perturbation mechanism in a federated matrix factorization algorithm for implicit feedback.
- In Chapter 6, we present the detailed architecture of our untraceable payment system. We build on the preliminaries of cryptographic primitives in Chapter 2 and show how those primitives are applied to our context.
- Finally, in Chapter 7, we discuss the main contributions of this thesis, highlighting the problems we tackled and opportunities we saw that would set the stage for future work.

Chapter 2

Background

This chapter explores the field of data privacy in e-commerce systems. We begin by looking at various state-of-the-art technologies used to accomplish essential tasks such as making recommendations and completing a fair exchange of goods and services. Next, we explore the importance of data privacy in e-commerce systems. We then introduce privacy-preserving technologies such as Cryptographic Primitives, Differential Privacy (DP) and Federated Learning and explain how these technologies are used to protect a user's privacy. We also give an overview of several significant works that have used these methods as privacy protection models in the context of this thesis.

2.1 E-Commerce Systems

An e-commerce system generally comprises various sub-information systems providing services for successfully executing a fair exchange of products and services through digital means. A critical aspect of an e-commerce system is personalisation. It has increasingly become a crucial component in most popular e-commerce systems such as Amazon and E-bay as it offers numerous benefits. Through personalisation, e-commerce systems can attract new customers and even aid existing customers in finding a new product or service that can fulfil their needs. They use recommendation systems to effectively infer meaningful information from customers' data and offer a personalised experience. Another critical aspect of an e-commerce system is the fair exchange of goods and services through efficient payment methods. Since payment is an integral part of

e-commerce systems, it has created new financial needs that traditional payment systems cannot fulfil. For example, most users opt to pay through mobile devices instead of card payment for face-to-face payment scenarios. Hence, several Third-Party Service Providers (TPSPs) are interested in offering various payment services to customers and merchants. This section gives an overview of different state-of-the-art recommendation systems and TPSP-based payment systems.

2.1.1 Introduction to Recommendation Systems

Recommendation systems provide users with recommendations based on their preferences. They have become a valuable tool to overcome information overload in so many domains such as e-commerce, healthcare, entertainment, etc. Recommendation systems rely on a user's historical information to accurately represent their interest profile. For example, Amazon uses information such as transaction history, views, search history etc., in their recommendation system to recommend items to users that they have not purchased yet (Smith and Linden, 2017). Assume A is a set of m users and B is a set of n items in a given e-commerce system. A recommendation system helps the TPSP to select and rank a subset of items from B, which has not been seen by a given user from the set of user A based on perceived relevance.

Most recommendation systems use explicit and implicit feedback to predict the relevance of an item for a user. Explicit feedback contains information given by users directly on items they have already purchased. Ratings and reviews are good examples of explicit feedback. Implicit feedback is drawn from the actions of users on items such as views, clicks or add-to basket. Recommendation systems can be classified into three categories based on what information and method are used to predict the relevance of an item for a user: Collaborative Filtering (CF), Content-based Filtering (CB) and Hybrid Systems (HS) (Adomavicius and Tuzhilin, 2005). CF-based recommendation systems use explicit or implicit feedback given by the users on the items, whereas CB-based recommendation systems use item features (e.g. movie genre) to make recommendations. HS-based recommendation systems combine different recommendation models and various types of information to find relevance. For example, some HS recommendation systems use both explicit feedback and item features to produce recommendations (Adomavicius and Tuzhilin, 2005).

Table 1: Example Rating Matrix

	Item 1	Item 2	Item 3	Item 4
User 1	5	3	-	1
User 2	4	-	-	1
User 3	1	1	-	5

Matrix Factorization

CF-based recommendation system that uses explicit feedback utilises a user-item rating matrix consisting of ratings of m users on n items to predict a user's preferences. User-based CF, item-based CF and model-based CF are three main methods used in a CF-based recommendation systems (Sarwar et al., 2001; Papagelis and Plexousakis, 2005; Aggarwal, 2016). User-based and item-based CF algorithms predict a user's preferences using either user or item similarities. K-Nearest Neighbour (KNN) is a popular approach used in user-based and item-based CF algorithms. Matrix Factorization (MF) is a popular model-based approach that aims to reduce the dimensionality of the rating matrix while discovering potential features. It reduces the dimensionality of the rating matrix by decomposing it into two matrices of lower dimensions. Many E-commerce systems prefer MF over other algorithms due to its higher predictive accuracy and computational scalability. MF has become the most popular method in latent factor-based recommendation systems after gaining popularity through the Netflix Prize competition (Bennett et al., 2007).

In an e-commerce system, we have a set of users and items. If the given users have rated some items, then using an MF-based recommendation model we intend to predict the ratings on items that users have not rated yet. The existing ratings can be represented using a rating matrix R that contains ratings of m users over n items that act as the input for the MF algorithm. Assume we have 3 users and 4 items, and the rating scale ranges from 1 to 5. Table. 1 shows how the rating matrix R will look like in such cases. The hyphen represents the items the user has not rated yet, and the MF-based recommendation model is used to predict these missing ratings.

MF-based recommendation models assume that some common latent features determine a user's rating on an item. Hence, if the model can discover these latent features, then the missing ratings can be predicted. It also assumes that the number of latent features would be smaller than the number of items and users. Let there be a k number of latent features. So the task is to find two

matrices $U_{m\times k}$ and $V_{n\times k}$ such that their product can approximate the rating matrix R:

$$R \approx U \times V^T = \hat{R} \tag{2.1}$$

Figure. 1 illustrates the intuition behind MF-based recommendation systems.

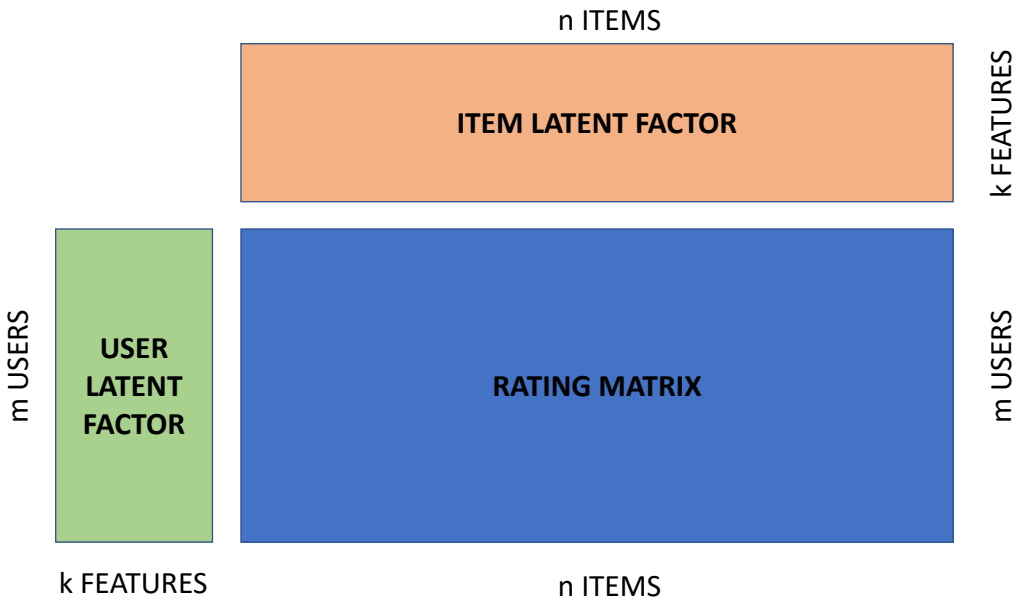


Figure 1: Decomposition of Rating Matrix into User/Item Latent Factor Matrices

Each element r_{ij} in the rating matrix indicates the rating of user $i \in \{1, 2, \cdots, m\}$ on item $j \in \{1, 2, \cdots, n\}$. An MF algorithm factorises the given rating matrix R into two latent matrices: U (user latent factor matrix) and V (item latent factor matrix) using the observed values. MF algorithm then predicts the missing rating by modelling the interactions between users and items as the inner product of latent factor spaces.

MF obtains the user and item latent matrices by minimising the squared error for all known ratings in the rating matrix.

$$\min_{U,V} \sum_{r_{ij}\in R} (r_{ij} - u_i^T v_j)^2, \tag{2.2}$$

where r_{ij} is the rating given by user i on item j. In Eq. (2.3), u_i represents the relationship between user i and the latent features k in the user latent matrix U. Similarly, v_j represents the relationship between item j and the latent features k in the item latent matrix V. Hence, Eq. (2.3)

can be rewritten as:

$$\min_{U,V} \sum_{r_{ij} \in R} (r_{ij} - \sum_{k=1}^{K} u_{ik} v_{kj})^2, \tag{2.3}$$

Despite the effectiveness of the objective function given by Eq. (2.3), it is proven that the rating prediction can be further improved when user and item biases are introduced (Koren et al., 2009). Hence the objective function can be revised as follows:

$$\min_{U,V} \sum_{r_{ij} \in R} (r_{ij} - u_i^T v_j)^2 + \lambda(||u_i||^2 + ||v_j||^2), \tag{2.4}$$

where λ is the regularisation parameter used to avoid over-fitting of the model. This non-convex optimisation problem given by Eq. (2.4) can be solved using either Stochastic Gradient Descent (SGD) (Koren et al., 2009) or Alternating Least Squares (ALS) (Hastie et al., 2015). In this thesis, we use SGD as the optimisation problem solver. An SGD algorithm iterates through all the ratings in a given training dataset and computes the prediction error e_{ij} as:

$$e_{ij} = r_{ij} - u_i^T v_j. \tag{2.5}$$

SGD updates the user and item latent factors as follows:

$$u_i^{t+1} = u_i^t + \gamma(e_{ij} v_j^t - \lambda u_i^t), \tag{2.6}$$

$$v_j^{t+1} = v_j^t + \gamma(e_{ij} u_i^t - \lambda v_j^t), \tag{2.7}$$

where γ is the rate used to minimise the error, usually referred to as the learning rate. The update procedure of latent factors using the SGD algorithm is inherently sequential. The latent factor matrices are updated using the values obtained in the previous iteration. The prediction error e_{ij} also changes in each iteration due to the change in latent factor matrices. Algorithm 1 explains the steps involved in the SGD algorithm.

After obtaining the latent factors, MF predicts the missing rating $\hat{r_{ij}}$ of a user i on an item j using

Algorithm 1 SGD for Matrix Factorization

1: **Input Rating matrix (R), Randomly initialised user and item factor matrices (U^0 and V^0) and λ**
2: **Output converged U^* and V^***
3: **while** not converged **do**:
4: **for** each rating r_{ij} in R **do**:
5: Update u_i using Eq. (2.6)
6: Update v_j using Eq. (2.7)
7: **end for**
8: **end while**

the dot product of the corresponding user and item latent column vectors:

$$\hat{r_{ij}} = u_i^T v_j. \tag{2.8}$$

More works have investigated different approaches to enhance the predictive accuracy of MF, such as incorporating MF with the KNN algorithm (Koren, 2008) and CB algorithms (Van den Oord et al., 2013). However, these works primarily rely on explicit feedback such as ratings to produce recommendations. Thus the performance of such systems relies on how much explicit feedback data they can gather from users. Since explicit feedback data is not always available in many applications, more and more TPSPs are inclined toward gathering implicit feedback from users when they interact with an item, e.g. purchase history. Even though TPSPs can collect implicit feedback much more easily than explicit feedback, capturing users' underlying intentions and preferences is more challenging, as implicit feedback does not directly reflect user satisfaction due to a lack of more explicit positive and negative feedback. For example, let's assume that the user-item interaction matrix Y contains m number of users and n number of items. Each element y_{ij} in the interaction matrix indicates whether a user $i \in \{1, 2, \cdots, m\}$ interacted an item $j \in \{1, 2, \cdots, n\}$. It can be defined as:

$$y_{ij} = \begin{cases} 1, & \text{if interaction between user } i \text{ and item } j \text{ exists,} \\ 0, & \text{if otherwise.} \end{cases}$$

In this interaction matrix, the value 1 means an interaction between user i and item j exists. However, it does not represent whether user i likes the item j. Similarly, the value 0 does not indicate negative feedback. Modelling all the missing data as negative feedback is one approach

that can solve this problem of insufficient negative feedback in the interaction matrix (Hu et al., 2008). More and more researchers are interested in investigating and solving utility-related issues in MF algorithms that use explicit or implicit feedback.

Content-based Recommendation Systems

In contrast to CF-based recommendation systems, a content-based recommendation system recommends items to users based on the correlation between the features of the items (e.g. genre of movies) and the preferences of users (e.g. ratings). The vector space method is commonly used to implement this strategy in content-based recommendation systems. The vector space method identifies similar items using the information given in the metadata of the items and uses techniques similar to TF-IDF to measure the similarity. This section explains how TF-IDF is used to produce recommendations in a content-based recommendation system.

Let's assume N is the total number of documents which contains metadata of each item, N_x is the total number of words in a given metadata document d_x and $N_{y,x}$ is the number of times a word y appeared in the document d_x. The term-frequency, $TF_{y,x}$ is defined as:

$$TF_{y,x} = \frac{N_{y,x}}{N_x}.$$

However, repeated words are not helpful in finding similar items. Hence, we utilise Inverse Document Frequency (IDF) which is given as:

$$IDF = log\frac{N}{n_y},$$

where n_y represents the number of documents that contains word y. Using TF and IDF we get the weight of the term y in document x as:

$$w_{y,x} = TF_{y,x} \times IDF$$

Given metadata documents of two items d_{x1} and d_{x2}, their similarity can be then measured by:

$$Sim(d_{x1}, d_{x2}) = \frac{\sum_{y=1}^{n} w_{y,x1} w_{y,x2}}{\sqrt{\sum_{y=1}^{n} w_{y,x1}^2} \sqrt{\sum_{y=1}^{n} w_{y,x2}^2}}$$

Once the similarity matrix is obtained, the recommendation systems will recommend the items with higher similarity to the user.

Hybrid Recommendation Systems

One common area of work in recommendation systems is combining different recommendation algorithms to obtain better performance. Such an approach helps the researchers to use the strengths of one recommendation algorithm to tackle the weakness of another one. For example, CF-based recommendation systems predict a user's preference using explicit feedback such as ratings. Even though CF algorithms produce satisfactory recommendation accuracy to an extent, they build upon the presumption that ratings reflect a user's true preferences or the actual quality of an item. This presumption does not always correspond with real-world scenarios, as impugn users tend to give lower ratings and tolerant users tend to give higher ratings. Such hypotheses play a huge role when a customer decides whether an item is suitable or unsuitable to fulfil their needs (Raghavan et al., 2012). Besides, users who give similar ratings to an item may have experienced distinct degrees of satisfaction (Cheng et al., 2018). Additionally, CF algorithms often suffer from data sparseness problems due to a lack of ratings which result in low recommendation accuracy (Mobasher et al., 2007).

Many works (Paterek, 2007; Shaowen and Yong, 2017; Pal et al., 2017) have investigated HS approaches to improve the predictive performance of CF algorithms. Since CF algorithms rely mainly on ratings and data sparseness is becoming an issue that needs to be tackled to improve predictive accuracy, TPSPs have started incorporating sentiment analysis to address these issues. Sentiment analysis is a technique used to categorise text-based data to further understand users' attitudes and opinions in several domains. By combining sentiment analysis with a CF algorithm, TPSPs can use ratings and reviews to improve recommendation accuracy.

Figure.2 illustrates a hybrid recommendation system which utilises explicit feedback such as user reviews and ratings. This recommendation system takes the outputs of two recommendation models (sentiment analysis model and CF algorithm) and combines the result to produce a recommen-

dation list. Preethi *et al.* (Preethi et al., 2017) are the first ones to introduce using a sentiment analysis model based on recursive neural networks in cloud-based recommendation systems. Wang *et al.* (Wang et al., 2018) incorporates a CF recommendation system with a sentiment analysis model to obtain an optimised preliminary recommendation list and then use that list to produce a final recommendation list. In another work, researchers propose a recommendation system where the context of the user's comments is taken into consideration and used to produce recommendations (Osman et al., 2019). Such approaches are more suitable when ratings are sparse and aid immensely in increasing recommendation accuracy.

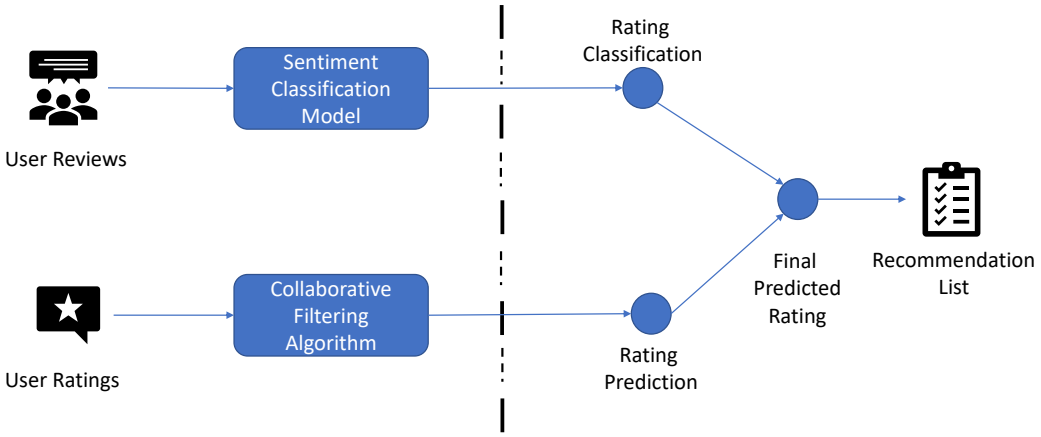


Figure 2: Hybrid Recommendation System with CF Algorithm and Sentiment Classification Model

2.1.2 Importance of Privacy in E-commerce systems

E-commerce systems heavily rely on data collected from users regarding their purchasing behaviours and preferences for personalisation and fair exchange purposes. Although e-commerce systems offer great conveniences for customers, TPSPs end up aggregating a large amount of user-sensitive information, inherently hampering user privacy and disclosing sensitive information to adversaries (Vakeel et al., 2017; Cen et al., 2019). Hence, user privacy has become an important issue to be tackled. Even though laws and regulations regarding privacy protection exist, they have proven inadequate and can not solve problems fundamentally.

Privacy Threats in Recommendation Systems

Recommendation systems are well known to help deal with the problem of information overload in e-commerce systems. They also enable TPSPs to predict users' preferences more accurately. However, they rely heavily on users' explicit/implicit feedback and past transaction history to generate these recommendations. Hence, they inherently raise legitimate privacy concerns as TPSPs gather a large amount of user data to predict users' purchasing behaviour. TPSPs can use such data aggregation and analytical practices to infer sensitive information regarding a user (Friedman and Schuster, 2010; Jeckmans et al., 2013; Lam et al., 2006). The rating data used by both memory-based and model-based recommendation systems poses a high privacy risk even if the user details are anonymized. Narayanan *et al.* (Narayanan and Shmatikov, 2008) demonstrates how such anonymized datasets can be deanonymized. Additionally, analysing users' historical ratings can also disclose sensitive information such as users' political preference, health-related information and sometimes even their sexual orientation (Singel, 2009; Kosinski et al., 2013; Weinsberg et al., 2012).

Even if a malicious third-party adversary has no direct access to the user's implicit/explicit feedback or transaction history data, it is still able to infer sensitive information about a user through the recommendations and public lists which contain information about a user's items of interest (Calandrino et al., 2011). These personal preferences expressed by the recommendation systems can be used as a quasi-identifier, mainly if users express unusual preferences, which leads to undesired re-identification of the user. However, such attacks are much more prevalent in memory-based CF systems than in their counterparts, model-based CF systems (Bilge et al., 2014). Yet, exposing model parameters in model-based CF systems can lead to privacy leakage (Chai et al., 2020).

Most users trust that the TPSP stores and keeps their information more securely and does not use it to infer any sensitive information about them. However, storing user information in a centralised server can lead to accidental or malicious disclosure of sensitive information (Foner, 1999). For example, a data breach at the TPSP can lead to identity theft, data being sold to spam advertisers or being shared on the dark web as potential targets for phishing attacks. There is also the risk of re-identification of a user using the information revealed by one e-commerce system (Sweeney, 2002). As a result, using this information e-commerce systems can offer incentives to attract users

away from their competitors. Hence, it is pivotal for the TPSP to protect users' privacy while providing suitable personalised recommendations.

Privacy Threats in Payment Systems

Customers increasingly find it harder to control who accesses their data in TPSP-based payment systems. Because of the exponential growth such payment systems have seen in recent years, TPSPs puts little effort into preserving a customer's privacy. TPSPs use credit/debit card numbers as a unique identifier for each customer in these payment systems. They can link multiple transactions to one customer using this identifier and build a comprehensive customer purchase profile. Such profiles are valuable for marketing and advertising purposes as they aid TPSPs in understanding customers' purchasing behaviours and expectations. Preibusch *et al.* (Preibusch et al., 2016) found that at least 50% of 881 online shopping sites shared customer information such as names, email addresses, phone numbers and item details with PayPal. Later, PayPal forwarded these details to Omniture, a third-party data analytical company, to infer usage statistics related to their users. Similarly, Ali Pay, which provides guaranteed secure third-party payment services for the shopping platform Taobao, also has been accused of performing unethical data aggregation and analytical activities (Reuters, 2018).

In most cases, customers request privacy in these systems not to protect themselves from government inspections but to protect themselves from unethical data analytical and aggregation practices of TPSP. For example, a customer should be able to choose not to reveal any information about a purchase to avoid any unpleasant ramifications due to the nature of the purchase. However, in today's popular TPSP-based payment systems, such desires are underappreciated. In 2020, Facebook announced its payment system based on cryptocurrency associated with other financial and technological companies. Their announcement indicates that a customer's transaction account will not be related to their social media accounts. However, the digital wallet they introduced as part of this project -Calibra- will be integrated with the Facebook marketplace and WhatsApp messenger applications leading to questions about their intentions regarding the aggregated data. The insights derived from such aggregated data would offer an incomparably substantial potential for monetising users' sensitive information. Even though customers and merchants can benefit from insights derived from their data, there is no denying that ensuring customer transactions are untraceable

by the TPSP is becoming an essential privacy and security requirement for TPSP-based payment systems.

2.2 Privacy Protection Technologies

A privacy-enabled e-commerce system offers changes to its traditional framework and algorithms to protect users' privacy. Specifically, such systems introduce changes to processes related to user data aggregation, data storage and data processing (Spiekermann and Cranor, 2008). Several techniques have been identified as framework-based strategies to enhance user privacy in both recommendation and TPSP-based payment systems. This section discusses three prominent privacy-enhancing technologies: Cryptographic Primitives, Differential Privacy, and Federated Learning.

2.2.1 Cryptographic Primitives

Cryptographic primitives are often used to limit the information revealed by distributed computation. Several cryptographic primitives such as blind signature, zero-knowledge proof, certificateless signature schemes and group signatures have been used to protect users' privacy in TPSP-based payment systems. This section briefly explains the cryptographic primitives used in this thesis.

Blind Signature

Blind signatures are used to provide anonymity for users in privacy-preserving applications such as e-voting, e-payment, and even to offer anonymous identities to users. They allow the users to obtain a signature on a message without revealing either the message or the resulting signature to the signer (Chaum, 1983). In return, the user cannot produce the signature of the signer without interacting with them (unforgeability). One of the primary applications of blind signature is anonymous identities (Baldimtsi and Lysyanskaya, 2013) where users can obtain anonymous identities while revealing only a few pieces of information about themselves. Another application of blind signature is e-voting (Kumar et al., 2017) where voters can cast their authorised votes without revealing their identities to the relevant authorities. In this thesis, we explore the usage of blind signatures in preserving a customer's privacy in TPSP-based payment systems. Fig.3

illustrates how a simplified blind signature scheme works.

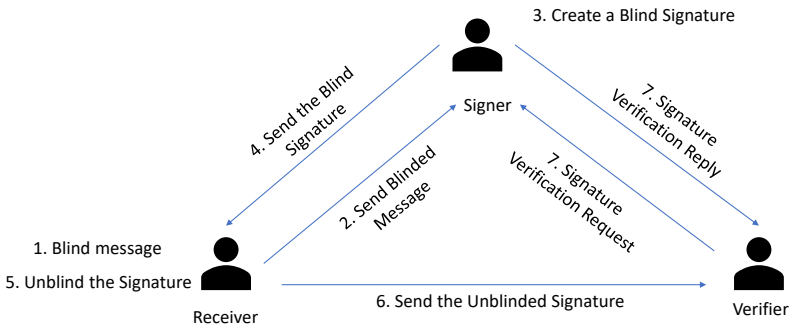


Figure 3: A Simplified Blind Signature Scheme

Signer is the role played by an entity that issues the signatures. Signer can be either a certification authority or a TPSP. The receiver requests a blind signature from the signer. Before requesting, the receiver blinds the message so that the signer cannot see what the actual message is about. Then the receiver sends the blinded message to the signer requesting a blind signature. Signer issues the blind signature without seeing the content of the message. After receiving the blind signature, the receiver unblinds the signature and obtains the actual signature on the message. After unblinding, the final verification stage of the blind signature scheme is identical to that of the regular digital signature scheme. In our work, we use an ID-based blind signature (Zhang and Kim, 2003) which uses the identity of the signer to produce the blind signature.

Let P be the generator of an additive cyclic group $G1$ whose order is prime q. G_2 is a multiplicative cyclic group of the same order q. The bilinear pairing e is given by $e : G1 \times G1 \rightarrow G2$. Let the signer's identity be ID, the master secret key s is known only to the signer and $H1, H2$ are two hash functions chosen by the signer. The signer sets the system parameter $P_{pub} = sP$. The public key of signer is given as $QID = H2(ID)$ and private key is given as $SID = sQID$. The following description details the blind signature algorithm:

- Signer shares the system parameters $G1, G2, e, q, P, P_{pub}, H1$ and $H2$ with each registered user and keeps the master key s private.
- Signer chooses a random number $r \in \mathbb{Z}_{\text{||}}$ and computes a commitment value $U = rQ_{ID}$. Signer then sends this commitment value U to the receiver.

- Blinding the message: Receiver then chooses blinding factors $\alpha, \beta \in \mathbb{Z}_{\amalg}$ and computes $U' = \alpha U + \alpha\beta Q_{ID}$ and a blind message $h = \alpha^{-1} H1(m, U') + \beta$ where m is the message that should not be revealed to the signer. The blind message h is then sent to the signer and U' is stored locally.
- Blind Signing: Signer then computes $V = (r + h)S_{ID}$ and sends V back to the receiver.
- Unblinding Signature: Receiver computes $V' = \alpha V$ and obtains the signer's signature on the message m. The signature would be (U', V').
- Verification of Signature: Verifier can accept the signature if and only if:

$$e(V', P) = e(U' + H1(m, U')Q_{ID}, P_{pub}).$$

Partially Blind Signature

Unlike blind signature schemes, partially blind signatures allow the signer or receiver to include some common information during the signature requests or issuance stage. For example, if an e-payment system uses a partially blind signature, the receiver can include information such as the date with the blinded message. Even though some information is revealed, the partially blind signature still preserves the unlinkability property desired by applications to protect user anonymity. Abe and Fujiski (Abe and Fujisaki, 1996) are the first ones to propose a partially blind signature scheme based on RSA (Rivest, Shamir, and Adleman). In our work, we use an RSA-based partially blind signature (Chien et al., 2001) which incurs low computation cost at the user-side compared to other schemes. This partially blind signature scheme consists of four phases:

- Initialisation: The signer generates the required information to successfully execute the partially blind signature and publishes them to the registered users.
- Signature Request: The receiver prepares the common information and the blind message.
- Signing: Signer signs the blinded message and sends it back to the receiver.
- Extraction: Receiver unblinds the signature.
- Verification: Verifier can verify whether the signature belongs to a respective signer.

The detailed description of each phase is as given below:

- Initialisation
  - Signer chooses two large prime numbers p and q and computes the modulus $n = p.q$, totient function $\varphi(n) = (p-1)(q-1)$, private exponent d in such way $e.d = 1 \; mod \; \varphi(n)$.
  - Signer then publishes (e, n) as their public key and keeps (d, p, q) as private.
  - Signer also publishes a chosen hashing algorithm $H3$.
- Signature Request
  - Receiver prepared the message m and the common information a to be used in this phase.
  - Receiver randomly chooses two numbers r and u where $r, u \in \mathbb{Z}_{\kappa}$ and computes blind message Y as follows:

$$Y = r^e H3(m)(u^2 + 1) \; mod \; n.$$

  - Receiver sends (a, Y) to the signer.
  - Signer randomly choose a positive integer x where $x < n$ and sends it to the receiver.
  - After receiving x, the receiver randomly generate another number r' and computes $b = r.r'$.
  - Then the receiver computers $\beta = b^e(u - x) \; mod \; n$ and sends β to the signer.
- Signing
  - After receiving β from the receiver, signer computes $\beta^{-1} \; mod \; n$ and computes t as follows:

$$t = H3(a)^d (Y(x^2 + 1)\beta^{-2})^{2d} \; mod \; n.$$

  - The signer then sends (β^{-1}, t) to the receiver.
- Unblinding Signature

- Upon receiving (β^{-1}, t), the receiver acquires the signature of the signer on the message m by computing:

$$c = (ux + 1).\beta^{-1}.b^e \; mod \; n = (ux + 1)(u - x)^{-1} \; mod \; n,$$

and

$$s = t.r^2.r'^4 \; mod \; n.$$

The receiver uses (a, c, s) as the signature on the message.

- Signature Verification
  - Verifier can check whether the signature is valid by performing:

$$s^e = H3(a)H3(m)^2(c^2 + 1)^2 \; mod \; n.$$

Zero Knowledge Proof

Zero-Knowledge Proof (ZKP) is an important cryptographic primitive used in many applications to provide anonymity to users. The main concept behind ZKP works as follows: The prover has some secret information (e.g. password or a secret key). They want to prove that they own this confidential information to the verifier without revealing the message to them. If the prover and the verifier comply with the ZKP protocol and proof holds, then the prover is considered credible. However, if the verification of the proof fails, then the prover is deemed to be counterfeit. The verifier cannot learn about the secret information during the execution of the ZKP protocol. In this thesis, we modify the TinyZKP (Ma et al., 2014), a lightweight authentication mechanism, to execute a mutual authentication protocol between a customer and a merchant. The following description summarizes the TinyZKP algorithm:

- A centralised server generates security parameters p and q which are kept private and $N = p.q$ is announced publicly.
- The server generates k number of public $(V_{m,1}, V_{m,2}....V_{m,k})$ and private $(S_{m,1}, S_{m,2}....S_{m,k})$ keys for each prover as follows:
  - Generation of private keys: Randomly generate an integer $S_{m,1}$ such that $1 \leq S_{m,1} \leq$

$N-1$

$$S_{m,j} = S_{m,1} - j + 1 \ (2 \leq j \leq k).$$

– Generation of public keys:

$$V_{m,j} = \frac{1}{S_{m,j}^2} mod\ N \ (1 \leq j \leq k).$$

Public keys $V_{m,j}$ are then send to the verifier.

- The verifier then generates a challenge $M_{chall} : (e_1, e_2, ...e_k)$ where $e_k = 0\ or\ 1$ and $k = 1, 2, ..., 20$. The verifier then sends the challenge (M_{chall}) and a timestamp T_1 to the prover.

- After receiving the challenge, the prover selects a random number r such that $1 \leq r \leq N-1$ and computes X_m and Y_m as follows:

$$X_m = r^2\ mod\ N,$$

$$Y_m = r \prod_{j=1}^{k} S_{m,j}^{e_j}\ mod\ N \ (1 \leq j \leq k).$$

- The hashed value $H(X_m)$ is computed and the prover sends $H(X_m)$, Y_m and a timestamp T_2 to the verifier as the challenge reply.

- The verifier initially verifies whether $\Delta t > T$ where $\Delta t = T_2 - T_1$ and T is the threshold response time set by the verifier to indicate before when the challenge reply has to be sent by the prover.

- If the verification is successful, the verifier computes X'_m as follows:

$$X'_m = Y_m^2 \prod_{j=1}^{k} V_{m,j}^{e_j}\ mod\ N \ (1 \leq j \leq k).$$

If $H(X'_m)$ is equivalent to $H(X_m)$ then the prover can prove to the verifier that they indeed have the knowledge of the secret information (e.g. Authentication details).

2.2.2 Differential Privacy

Most applications use privacy protection models based on obfuscation (Parameswaran and Blough, 2008) and perturbation (Jain and Bhandare, 2011) techniques. These approaches introduce random noise to the data. Yet, the magnitude of noise added using these methods cannot be calibrated easily. Hence, the applications that use such approaches suffer from the low utility of data. DP-based (Dwork, 2008) mechanisms are another popular perturbation approach used in privacy protection models to tackle this problem. DP-based methods are proven to be a stronger solution for privacy protection in various applications compared to other perturbation methods. DP provides an information-theoretic guarantee of strong privacy protection regardless of how much knowledge the adversary possesses. Unlike other perturbation approaches, in DP, the calibration of noise depends on the sensitivity of the query and the level of privacy offered to the user. DP-based privacy protection models are relevant in settings where the TPSP is trusted and aggregates users' original data.

DP formalises the intuition of whether a user's data is present in the database. It compares the probability distribution of possible outputs of an algorithm when a user's data is present in a given database, to the probability distribution of the same possible set of outputs when the user's data is not present in the database. If these two distributions are almost the same, then we can negate the effect of a user's data causing any possible output. Such comparison guarantees that a user's participation in a database does not cause any privacy risk to the user. Let D be the dataset of m attributes and n records. Two datasets D and D' are neighbouring datasets if they differ in at most one row, corresponding to one individual's data. f is a query function that maps the dataset D to a real number $f : D \rightarrow R$. M is a differentially private mechanism used to hide the difference between the output produced by query f on datasets D and D'. The maximum difference in the outputs of the query f is defined as the sensitivity Δf. The definition of differential privacy can be formalised as (Dwork, 2008):

Definition 1. *A randomised mechanism M satisfies ε-differential privacy if for any adjacent datasets D and D', and any subset S of all possible outputs, we have the following inequality:*

$$Pr[M(D) \in S] \leqslant e^{\varepsilon} \times Pr[M(D') \in S],$$

where ε is the privacy budget.

From the definition, it is clear that DP is a condition that bounds the ratio between the probability distributions of randomised algorithm M on datasets D and D'. The privacy budget ε controls this ratio. The closer ε to zero, the more similar the two distributions are. The smaller the value of the privacy budget ε, the lower the confidence the adversary has in distinguishing whether dataset D or D' produced the output. This indicates that the privacy loss is less and privacy protection is high. Hence, DP provides a higher degree of privacy protection for lower values of privacy budget ε.

The intuition behind the randomised mechanism M can be shown as :

$$M(D) = f(D) + noise,$$

where f is a deterministic query function. For example, f can be a counting query. The mechanism M is private if neighbouring datasets produce nearby distributions on the outputs. Hence, without randomisation, the distributions will be distinct, and the adversary can infer whether an individual has a pronounced effect on the observed outputs.

Sensitivity

The randomised mechanism that satisfies DP decides how much noise perturbation is required based on sensitivity. Sensitivity derives the maximum difference between the results of a query function on two adjacent datasets, D and D', which are inputs to a differential private randomised algorithm. The sensitivity of a query function can be defined as follows:

Definition 2. *Given a deterministic function $f : D \to \mathbb{R}$, the sensitivity of f, Δf, can be defined as:*

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|.$$

Here f is a deterministic query function which maps dataset D to real numbers. For example, the deterministic function can be the percentage of smokers in a given dataset D. The sensitivity is calculated to identify how much a single individual's data can change the output of function f. If a user's data can change the output of the query function by a lot, then more random noise needs

to be introduced to hide their participation.

Laplace Mechanism

The Laplace mechanism is one of the prominent differentially private mechanisms. It adds random noise drawn from the Laplace distribution of mean 0 and variance b to preserve ε-differential privacy. The scale parameter b controls the width of the Laplace distribution. If Δf is the sensitivity of the query f and ε is the privacy budget, then the scale parameter b_{lap} of Laplace distribution can be determined as follows:

$$b_{lap} = \frac{\Delta f}{\varepsilon}.$$

Hence, the width of the Laplace distribution is dependent on sensitivity Δf and privacy budget ε. We use the notation $Lap(0, \frac{\Delta f}{\varepsilon})$ to indicate the Laplace mechanism.

Definition 3. *Given a query $f : D \to \mathbb{R}$, the Laplace mechanism M satisfies ε-differential privacy if:*

$$M(D) = f(D) + Lap(0, \frac{\Delta f}{\varepsilon}).$$

Local Differential Privacy

The major part of the DP-based works has focused on the centralised model. In this model, the user data is aggregated by a trusted TPSP who publishes the statistical aggregates of a differentially private computation. The TPSP has access to original user data such as browsing history, ratings, location history, etc., in such settings. The TPSP can use these data to learn sensitive information about the user. Additionally, even if the user trusts the TPSP with their data, the possession of large amounts of sensitive data poses a major security risk. Accidental data breaches of such personal data lead to severe consequences and privacy violations. Recently many works have started using the local model (Duchi et al., 2013), where each user perturbs their data locally on their own devices and sends the perturbed data to the TPSP. The requirement is that the perturbation algorithm satisfies the following definition:

Definition 4. *A randomised mechanism M satisfies ε-local differential privacy if for all possible pairs of a user's individual data x, x' and any subset y of all possible outcomes, we have the*

following inequality:

$$Pr[M(x) \in y] \leqslant e^{\varepsilon} \times Pr[M(x') \in y].$$

Since the TPSP aggregates only the perturbed output, they can't infer any information about the actual data by observing the perturbed output even though they possess substantial background knowledge about the user. Hence, Local Differential Privacy (LDP) offers plausible deniability to users. Intuitively, LDP ensures that the TPSP cannot infer whether a user's input x or x' produces the output y with confidence. In this regard, LDP offers a more robust level of privacy protection than DP settings.

Properties of LDP

LDP satisfies properties such as sequential composition, parallel composition and post-processing (Xiong et al., 2020). These properties for LDP can be given as follow:

- Sequential Composition: If the user executes i number of perturbation algorithms in a sequence where each algorithm satisfies ε_i-LDP, then the whole system satisfies $\sum \varepsilon_i$-LDP.
- Parallel Composition: If the dataset is partitioned into groups and each group uses a ε-LDP perturbation algorithm, the whole system satisfies ε-LDP.
- Post Processing: Applying any process to perturbed outputs of a ε-LDP algorithm does not violate LDP principles and still guarantees privacy.

2.2.3 Federated Learning

Federated learning (McMahan et al., 2017) is a machine learning setup where each user in a centralised server-based system train a model locally on their devices under the supervision of a TPSP. The training data is kept decentralised during the training process to mitigate any privacy risks caused by the TPSP, which aggregates all the user data. Federated learning algorithms have received significant interest from both researchers and various TPSPs. Various tech giants have introduced federated learning-based data analytical practices in their applications. Google uses federated learning models in their Gboard mobile keyboard to improve query suggestions (Yang et al., 2018) and in Android messages to improve predictive suggestions (Google, 2019). This section describes the essential characteristics of a federated learning setup and highlights the steps

involved in a federated matrix factorization algorithm.

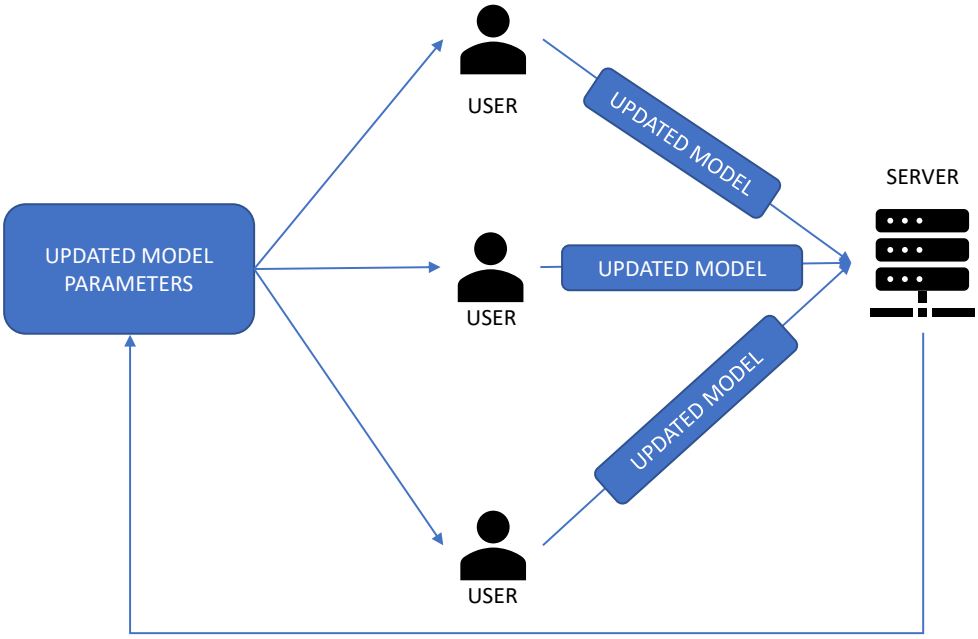


Figure 4: Federated Learning Model with a Centralised Server

Fig.4 illustrates the general architecture of a federated learning model. At a higher level, the typical workflow of a federated learning model involves:

- Problem Identification: Identifying a domain where the TPSP can deploy a federated learning model to preserve a user's privacy (E.g. Recommendation Systems)
- Data Aggregation at the User-Side: The relevant apps running at the user-side stores the necessary training data locally without sending it to the TPSP (e.g. e-commerce app storing user ratings on items locally)
- Federated model training: Users download the global model and hyper-parameters from the server and train the model locally using their training data. After the first round of training, they send the updated model parameters to the server. The server accumulates the model parameters from all the users and updates the global model accordingly. This training process is repeated for several rounds until the model reaches its convergence point.
- Federated model evaluation: After completing the training phase, the model is analysed based on various metrics either using the standard data available with the TPSP or by em-

ploying a federated evaluation process where users evaluate the model using their data.

- Deployment: Once a good model is trained and evaluated, the TPSP initiates the standard model deployment process.

Implementation of the above process requires users to send updated model parameters back to the server during each round. This step becomes a bottleneck if the model's size is large due to different factors. For example, an optimisation problem-solving algorithm like SGD, which uses a large partitioned dataset, requires low-latency and high-throughput connections in the federated learning setting. However, standard user devices have high latency and low throughput connections. Hence, many works have proposed effective approaches to reduce communication rounds and improve model upload speed (McMahan et al., 2017; Nishio and Yonetani, 2019; Smith et al., 2017; Jiang and Ying, 2020).

Apart from communication costs, tackling security issues is another problem that has to be addressed in federated learning models. A backdoor attack was designed to demonstrate that the federated learning models are prone to data poisoning attacks (Bagdasaryan et al., 2020). Another work (Bhagoji et al., 2019) shows how a fault training data could affect federated learning models and generate a wrong prediction. Existing works that aim to prevent such attacks in federated learning settings are not entirely valid, and more researchers are looking into mitigating these types of attacks.

In federated matrix factorization, users compute item latent factor stochastic gradients using their rating data and then send the stochastic gradients (instead of the ratings) to the TPSP. In regular matrix factorization algorithm the latent factors U and V are obtained by minimising the objective function given by Eq. (2.4). SGD iteratively updates U and V using Eq. (2.6) and 2.7. This iterative process is decomposed into two parts in a federated matrix factorization model. Eq. (2.6) is used at the user-side and Eq. (2.7) is executed at the server-side. Algorithm 2 summarises the steps involved in federated matrix factorization.

Algorithm 2 Federated Matrix Factorization

Input: Server initialises item latent factor (V^0)
Input: Each user initialises their own user latent factor (u_i^0)
Output converged U^* and V^*
Server publishes the item latent factor (V^0) for all users to download
User Update
    Update u_i using Eq. (2.6)
    Computer Item latent factor stochastic gradient
    Send the stochastic gradient to the server
Server Update
    Receive the stochastic gradient from each user
    Update v_j using Eq. (2.7)

2.3 Existing Work

2.3.1 Untraceable Payment Systems

TPSP-owned payment systems allow customers to pay for their transactions in a fraction of a second. However, they do not protect a customer's privacy in the face of a TPSP. Bitcoin is a widely spoken solution to ensure the untraceability of transactions in a TPSP-based payment system. Bitcoin operates over a peer-to-peer overlay network without interference from a TPSP which can be either a government or a non-government agency (Nakamoto, 2008). It is proven to be cryptographically strong, and users can make purchases without worrying about anyone tracking their transaction details. However, in recent years Bitcoin has been used as a preferred payment method in illegal activities such as drugs and human trafficking (Foley et al., 2019). Several works (Karame et al., 2012; Heilman et al., 2015; Reid and Harrigan, 2013) have investigated the privacy flaws caused due to the uncontrollable and decentralised nature of Bitcoin payment systems. The shortcomings we have observed in Bitcoin payment systems emphasise the need for a TPSP to regulate transactions between customers and merchants.

Cryptographic primitives are valuable for designing a centralised untraceable payment system where transaction details involving a customer and merchant are not revealed to the TPSP. However, the customer and merchant should still be able to successfully perform a fair exchange of goods and services with the assistance of the TPSP. Centralised untraceable payment systems are classified as online and offline systems based on the interaction between the customer, merchant and the TPSP. In online untraceable payment systems (Sai Anand and Madhavan, 2000; Martínez-Peláez and Rico-Novella, 2006), the TPSP has to verify the validity of the payment token used

by the customer to pay the merchant before the end of a transaction to prevent the customer from double-spending their payment token. Hence, the TPSP must constantly stay online during all the transactions to validate each payment token to detect double-spending attacks.

Offline payment systems (Anderson et al., 1996; Baseri et al., 2013; Deya et al., 2012) on the other hand, permit a customer to pay the merchant without requiring much involvement from a TPSP. Once the transaction completes, the merchant deposits the payment token obtained from the customer to the TPSP at a convenient time. However, there is a higher chance that a double-spending attack will be successful in offline payment systems if there is no prevention mechanism. Some offline payment systems (San and Sathitwiriyawong, 2016; Zhao et al., 2009) maintain untraceability by issuing an anonymous identity to each customer. In these systems, first, a customer purchases an anonymous pseudonym from another third-party anonymous identity provider or the TPSP. Then the customer uses these identities to pay the merchant. However, if it is the TPSP who issues these anonymous identities, they can still track a customer's transaction details. Additionally, these schemes require the customer to trust the third-party identity provider not to reveal the link between their actual identity and the pseudonym to the TPSP. Some offline payment systems (Deya et al., 2012; Wang, 2011) assume that merchants are honest and thus, do not have any preventive mechanism in place to stop a merchant from double-spending the payment token. Most online and offline untraceable payment systems assume that the TPSP is trustworthy and offers privacy protection against a malicious third-party adversary or a dishonest merchant.

Various cryptographic primitives have been introduced to offer untraceability in TPSP-based payment systems. Some works have offered untraceability in payment systems through blind signature schemes (Chaum, 1983; Baseri et al., 2013; Fan and Lei, 2002). Blind signature schemes allow a customer to obtain a validated payment token from the TPSP without revealing additional information. However, since the customer cannot disclose any information, multiple forms of payment tokens have to be issued to differentiate face values. Hence, these schemes incur more computational and storage complexity. Some works have introduced untraceable payment systems based on partially blind signatures to address this issue (Cao et al., 2005; Abe and Fujisaki, 1996). A partially blind signature allows the customer to reveal some information to the TPSP. However, these systems also add additional computational costs on the customer side and are not suitable for mobile payment systems. In another work (Zhang et al., 2015) zero-knowledge proof-based untrace-

able payment scheme is used to provide anonymity for the customer during transactions. Besides zero-knowledge proof, this system also uses group blind signatures to attain optimal anonymity. However, this scheme requires the customer to obtain a certificate from the group manager, which introduces more actors to the existing TPSP-based system architecture.

In summary, most of the works proposed in the literature tend to assume that the TPSP is trustworthy and allows them to link a payment token to its owner. They also fail to deal with double-spending attacks performed by a dishonest merchant. Even though some works have proposed preventive mechanisms to prevent double-spending attacks by fraudulent customers, they have failed to present a method that reveals the identity of the dishonest customer. Some of these payment systems also haven't considered the possibility of the merchant revealing the customer's identity to the TPSP. Additionally, most of these works add additional computational complexity to the customer side, making it impractical to implement them in a real-world scenario.

2.3.2 Privacy Protection in Recommendation Systems

As a prominent privacy-preserving model, DP has been widely used to protect a user's privacy in recommendation systems. Many DP-based recommendation systems assume a trusted TPSP who collects users' ratings and releases information related to users' preferences under the DP guarantee. McSherry and Mironov (McSherry and Mironov, 2009) are the first ones to integrate the DP-based privacy protection model with CF recommendation systems. In their method, TPSP is considered trustworthy. They build a covariance matrix using the user's original ratings that resemble the similarity between users. They then apply the Laplace mechanism to perturb the covariance matrix before predicting missing ratings. In their method, the trusted TPSP still has access to sensitive unperturbed data of a user. Yakut and Polat (Yakut and Polat, 2010) also introduce a DP-based recommendation system where the user's original ratings are stored at TPSP and then perturbed, which provides uncertainty over the user's actual ratings. This method also ensures that some user profiles containing fake ratings are included to further provide uncertainty over a user's preferences.

Even though DP-based recommendation models offer privacy protection to users from third-party adversaries, they enable TPSPs to collect the original ratings from users, which in return causes privacy concerns. Recently the attention of researchers has gradually shifted from DP to LDP as

users have become reluctant to allow a TPSP to collect their data. LDP is a stronger notion of a privacy model where each user perturbs their data locally, so a user's privacy is guaranteed even in the face of a malicious TPSP. Many applications have adopted LDP to deal with an untrustworthy TPSP. Erlingsson *et al.* (Erlingsson et al., 2014) propose the first LDP-based perturbation mechanism, RAPPOR, for crowd-sourcing data aggregation. Google uses this mechanism to collect users' Chrome usage statistics privately. RAPPOR uses a modified randomised response (RR) mechanism to independently perturb each bit of the data before sending it to the server for further analysis. However, high communication overhead during the data collection phase is a drawback found in this method.

Bassily *et al.* (Bassily et al., 2017) propose another LDP-based perturbation mechanism to address the issue found in RAPPOR. In their method, each user report one randomly chosen bit rather than reporting all the bits back to the TPSP using the succinct histogram (SH) mechanism. However, the SH perturbation mechanism is more suitable for simple numeric or categorical attributes but not appropriate for more complex data mining tasks. Qin *et al.* (Qin et al., 2016) also propose an algorithm for performing heavy hitters estimation under the guarantee of LDP. In another work, Wang *et al.* (Wang et al., 2017) propose another LDP algorithm for data aggregation and decoding. These methods can generate perturbed data and reconstruct the statistical characteristics simultaneously. Yet, they cannot recreate the cross-correlation relationship between data. To address this problem, Zhang *et al.* (Zhang et al., 2018) proposes an LDP algorithm that can reconstruct cross-correlation relationships among high-dimensional data. In their method, marginal tables are generated and then perturbed, and only the noisy marginal tables are sent to the TPSP.

Several works have investigated using LDP in CF recommendation systems. In their work, Liu *et al.* (Liu et al., 2015) propose a privacy-preserving recommendation system that uses a randomised perturbation mechanism. First, noise is added to users' ratings locally on the user side through a randomised perturbation method. Additionally, they add noise to the correlation computation method executed at the TPSP to further guarantee privacy. Even though this method includes added privacy protection, it incurs more predictive accuracy loss as noise is added during data aggregation and analytical stages. Meng *et al.* (Meng et al., 2018) address this problem by concentrating on perturbing ratings that are considered sensitive. They divide a user's historical ratings into sensitive and non-sensitive ratings. They use a large magnitude of noise to perturb

only the sensitive ratings, which reduces the amount of noise added to non-sensitive ratings during the data aggregation process. Hence, sensitive ratings receive better privacy protection, while the recommendation system achieves improved predictive accuracy. Still, the distinction between sensitive and non-sensitive ratings can vary from user to user and cannot be generalised to all users, which is a drawback of this method.

Shen and Jin (Shen and Jin, 2014) propose an instance-based admissible mechanism to perturb users' private data. They aim to hide users' preferences towards an item from an untrusted TPSP. However, this method can still reveal users' preferences towards an item category. Hua *et al.* (Hua et al., 2015) propose another LDP-based recommendation model where the TPSP uses LDP-based MF to compute the item latent factors and not the user latent factors. Subsequently, TPSP sends these item latent factors to the users to compute their user latent factors. Each user then sends the updated item latent factors back to the TPSP. This method requires the users to remain online during the whole MF process. Their proposed model adds additional communication and processing cost on the user side. Shin *et al.* (Shin et al., 2018) propose an LDP-based recommendation model where they use a randomised response mechanism to perturb the stochastic gradient and send the perturbed stochastic gradient back to the TPSP. This method incurs additional processing and communication overhead to the user-side the same as (Hua et al., 2015). In another work, Berlioz *et al.* (Berlioz et al., 2015) investigates the effect of rating perturbation in different stages of the recommendation process. They evaluate the role of input, gradient and output perturbation mechanisms on the final predictive accuracy.

To summarise, existing LDP-based recommendation systems suffer from low predictive accuracy and communication/computational overhead at the user side. Some proposed methods focus on frequency estimation, such as heavy hitter identification. These methods are unsuitable for CF because they focus only on a specific candidate set and cannot identify the correlation between users/items. Another group of works have concentrated on introducing a perturbation mechanism that adds noise only to ratings that are considered sensitive. However, such distinctions have to be made at the user level, and it would add more complexity to the existing recommendation system architecture.

2.3.3 Federated Learning-based Recommendation Systems

Implicit feedback-based recommendation systems have become popular in the past few years because they can achieve higher predictive accuracy than other conventional models. However, such success is primarily attributed to large-scale user behavioural data aggregation. These large-scale data aggregation has increased concerns over the intentions of a TPSP and their willingness to protect a user's privacy. Federated learning-based recommendation models offer a privacy solution where the TPSP does not need direct access to user behavioural data. Yet, they can still build and produce accurate recommendations for users. Training in a federated learning model consists of four key steps. First, the TPSP randomly selects a batch of users who are available for training and sends them the global parameters of the recommendation model. Each user trains the model locally using their data and the global model parameters shared by the TPSP. Finally, the TPSP accumulates the updated local parameters from each user and updates the global model. These training steps are iterated until the model meets the convergence requirements.

The Federated learning approach is suitable to train any gradient-based learning algorithms (Kendall et al., 2015; Yang et al., 2018) and even more ideal for MF-based recommendation systems. In federated matrix factorization (Chen et al., 2018), each user updates their latent factors locally and only sends the stochastic gradients of item latent factors to the TPSP. The TPSP updates the item latent factor using these stochastic gradients. However, researchers have proven that the TPSP can still learn about users' ratings on items through these stochastic gradients they receive (Chai et al., 2020). Hence, the researchers propose using homomorphic encryption-based cryptographic primitives to address this problem (Chai et al., 2020). In another work, researchers have proposed a generic federated matrix factorization algorithm in which a user's original rating and their latent vectors are stored locally, and only the stochastic gradients of the chosen low-sensitive items' latent vectors are sent to the TPSP to update the global model (Yang et al., 2021). Another work proposes a federated learning-based recommendation system in which they use a DP-based perturbation mechanism to perturb the stochastic gradients received from the users and also uses homomorphic encryption to communicate these gradients to the TPSP securely (Du et al., 2021). The global item latent factors are computed using the perturbed gradients in this method.

To summarise, federated learning is a relatively new concept in recommendation systems. In a federated matrix factorization system, users do not have to send their explicit or implicit feedback to

the TPSP. However, the TPSP can still predict users' preferences through distributed model training. Some works have identified that TPSPs can infer sensitive information through the stochastic gradients they receive from users even in a federated learning setting. Existing solutions primarily have evaluated the use of homomorphic encryption and multi-party-based computation to protect users' privacy. Such solutions increase the computational cost at the user side and do not protect users' privacy from a trustworthy TPSP. Additionally, when a DP-based perturbation mechanism is used to perturb the gradient, it still allows the TPSP to aggregate the original gradients. Moreover, most of these works concentrate only on explicit feedback and do not address the problems related to implicit feedback.

Chapter 3

LDP-based Collaborative Filtering Recommendation System

Recommendation systems rely heavily on the behavioural and preferential data (e.g. explicit feedback) of a user to produce accurate recommendations. However, these data aggregation and analytical practices of a Third-Party Service Provider (TPSP) cause privacy concerns among users. Hence, Local Differential Privacy (LDP) has attracted much attention as it can provide a strong privacy guarantee in a setting where TPSP is untrustworthy. Many researchers (Berlioz et al., 2015; Shin et al., 2018) have adopted LDP to protect the privacy of users in recommendation systems. Each user adds noise to their data locally in LDP-based privacy protection models and forwards the perturbed data to the TPSP. As the original data never leaves the user device, users are guaranteed plausible deniability. However, adopting LDP in recommendation systems causes low data utility for TPSP. Therefore, it is crucial to design an LDP-based recommendation system that provides strong privacy protection to users and simultaneously offers higher data utility to the TPSP.

Motivated by this, we propose an LDP-based recommendation system that perturbs the original ratings of a user within a predefined domain using the Bounded Laplace (BLP) mechanism. We then use a Mixture of Gaussian (MoG) model to estimate the aggregated noise at the TPSP to enhance the data utility. The main contributions of this chapter are listed below:

- We introduce BLP as an input rating perturbation mechanism to increase the recommenda-

tion accuracy of the LDP-based recommendation systems. To the best of our knowledge, we are the first to introduce BLP as the input data perturbation mechanism and to provide a sufficient condition for BLP to satisfy ε-local differential privacy in recommendation systems. We also empirically evaluate the BLP mechanism's role in enhancing predictive accuracy.

- The probability density function of BLP noise is conditional on the input rating matrix, unlike the Laplace mechanism, where there are no bounding constraints to restrict the noise samples. Hence we perform a theoretical analysis to identify and yield a noise distribution for the BLP mechanism. We derive a closed-form probability density function for noise drawn from BLP for a given dataset.

- We introduce a noise estimation component at TPSP to further increase the predictive accuracy of the recommendation system. Perturbation of each user's rating leads to higher predictive error, which increases linearly with the number of users and items. We adopt Matrix Factorization (MF) with a Mixture of Gaussian (MoG) to estimate the aggregated noise at TPSP and, at the same time to predict missing ratings. This novel approach tackles data utility issues found in LDP-based recommendation systems. We empirically evaluate the effect of MF with MoG in terms of achieving higher recommendation accuracy and show that the proposed LDP-based recommendation model outperforms the existing LDP-based recommendation models such as (Shin et al., 2018) and (Berlioz et al., 2015).

- Our approach causes much lower communication costs compared to existing LDP-based recommendation systems e.g.(Shin et al., 2018). Users only need to transmit each perturbed rating once to the TPSP in our proposed method. On the contrary, in other systems such as (Shin et al., 2018), the information exchange between a user and the TPSP continues for several iterations until the solution converges.

Our method protects users' privacy from an untrustworthy TPSP. However, we need to indicate some cautions and limitations related to this model. Firstly, we assume that each user sends a single rating to the TPSP at any given time, and this rating is independent of other users' ratings. We presume all the ratings are sensitive and essential in building a behavioural profile for a user. Hence, we perturb all users' ratings, causing heavy utility loss to the TPSP. We provide a solution to address this issue - combining a noise estimation model (MoG) with the recommendation al-

gorithm to sanitise the obfuscated data at the TPSP. Secondly, we only mask the rating scores of users to items, but not the set of items a user has rated. Exposure of user-item association to the TPSP can harm user privacy. However, hiding the relationship between users and items will further reduce data utility. We can apply our proposed privacy-enhanced recommendation system to an existing centralised recommendation model with satisfactory recommendation accuracy and low communication and computation costs. We will seek solutions to strengthen privacy protections further against the exposure of user-item linkage in Chapter 6.

In Table 2, we list notations frequently used in this chapter.

Table 2: LDP and MF Notations

Notation	Meaning
Pr	Probability
R	Original rating matrix
R^*	Perturbed rating matrix
N	BLP noise matrix
r_{ij} or r	Original rating
r^*_{ij} or r^*	Perturbed rating
n_{ij} or n	BLP noise
l	Minimum value in rating scale
u	Maximum value in rating scale
U	User Latent Factor Matrix
V	Item Latent Factor Matrix
u_i	Latent factors of user i in latent matrix U
v_j	Latent factors of item j in latent matrix V

3.1 Local Differential Privacy Recommendation with BLP and MoG

Our proposed recommendation model is applicable in a setting where the users are cautious about sharing sensitive information with an untrustworthy TPSP. Fig. 5 illustrates the proposed recommendation system. An LDP mechanism, BLP, perturbs users' actual ratings before sending them to the TPSP. Hence, the TPSP can only aggregate perturbed ratings from the users. At the TPSP, MF with MoG model estimates the noise added to the ratings and performs missing rating prediction. The post-processing property of LDP implies that further processing a perturbed output of a ε-differentially private mechanism does not cause any adverse effects on privacy protection (Dwork, 2008). Since LDP mechanisms are immune to post-processing, estimating noise at the

TPSP does not cause any additional privacy risk to users. We will describe each component of the system in detail in this section.

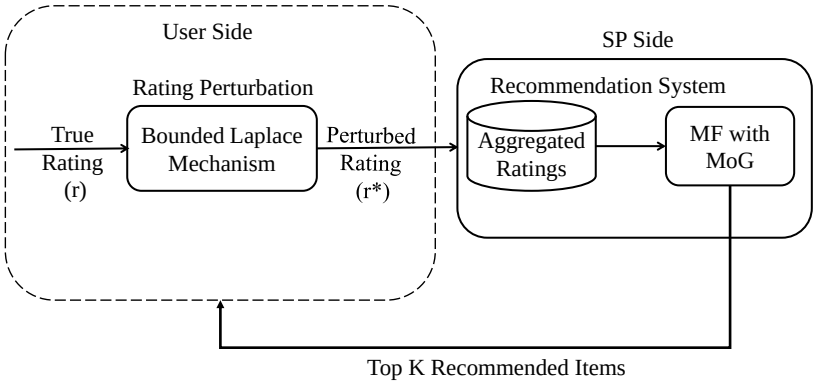


Figure 5: LDP-based MF Recommendation with MoG

3.1.1 LDP Rating Perturbation

Bounded Laplace Mechanism

The Laplace mechanism achieves ε-differential privacy by sampling random noise from $-\infty$ to ∞. The perturbed output thus falls within the domain of $-\infty$ to ∞. For example, the Laplace mechanism might produce a negative result as a perturbed output while perturbing a user rating. Although this negative output holds no physical meaning in terms of the rating scale, it is still a valid output of the Laplace mechanism. Such inconsistent perturbed ratings significantly affect MF-based recommendation systems' predictive accuracy.

We use BLP as an input perturbation mechanism to increase the predictive accuracy of LDP-based recommendation systems. The BLP mechanism continuously ignores off-limit values and sample noise for a given input rating until the perturbed rating falls within the predefined output domain. Given an input rating r, the BLP mechanism continuously samples noise from a Laplace distribution until the perturbed rating r^* falls within the predefined output domain, i.e. $l \leq r^* \leq u$, where l is the minimum and, u is the maximum value of the given rating scale.

Bounded Laplace mechanism can be defined using the probability density function (pdf) as below (Holohan et al., 2020):

Definition 5. *Given a domain interval of* (l, u)*, input* $r \in [l, u]$ *and the scale parameter* $b > 0$*,*

the Bounded Laplace mechanism W, is given by the conditional probability density function :

$$f_W(r^*) = \begin{cases} \frac{1}{C(r)} \frac{1}{2b} e^{-\frac{|r^* - r|}{b}}, & \text{if } r^* \in [l, u], \\ 0, & \text{if } r^* \notin [l, u], \end{cases}$$

where $C(r) = \int_l^u \frac{1}{2b} e^{-\frac{|r^* - r|}{b}} dr^*$ is a normalisation factor dependent on input r.

Lemma 1. *The normalization factor $C(r)$ is given by:*

$$C(r) = 1 - \frac{1}{2}\left(e^{-\frac{r-l}{b}} + e^{-\frac{u-r}{b}} \right).$$

Proof.

$$\begin{aligned} C(r) &= \int_l^u \frac{1}{2b} e^{-\frac{|r^* - r|}{b}} dr^* \\ &= \int_l^r \frac{1}{2b} e^{\frac{r^* - r}{b}} dr^* + \int_r^u \frac{1}{2b} e^{-\frac{r^* - r}{b}} dr^* \\ &= \frac{b}{2b} \left[e^{\frac{r^* - r}{b}} \right]_l^r + \frac{b}{2b} \left[-e^{-\frac{r^* - r}{b}} \right]_r^u \\ &= 1 - \frac{1}{2}\left(e^{-\frac{r-l}{b}} + e^{-\frac{u-r}{b}} \right). \end{aligned}$$

□

Assume that r and r' are a pair of possible inputs to a randomised mechanism and $r' = r + z$. We define $F(r, z)$ as:

$$F(r, z) = \frac{C(r + z)}{C(r)} e^{\frac{|r' - r|}{b}}.$$

Lemma 2. *Let $0 \leq z \leq \Delta f$, then,*

$$\max_{\substack{r, r' \in [l, u] \\ 0 \leq z \leq \Delta f}} F(r, z) = \frac{C(l + \Delta f)}{C(l)} e^{\frac{\Delta f}{b}}.$$

Proof. Assume r and r' are a pair of possible inputs of the BLP mechanism where $r' \geq r$ and $r' = r + z$. Let $0 \leq z \leq \Delta f$. In order to prove Lemma 2, we must first consider few other

properties concerning $C(r)$. First we find $\frac{\partial}{\partial z}F(r,z) \geq 0$ when $r+z \leq u$.

$$\begin{aligned}
\frac{\partial}{\partial z}F(r,z) &= \frac{1}{C(r)}\frac{\partial}{\partial z}\left(C(r+z)e^{\frac{z}{b}}\right) \\
&= \frac{1}{C(r)}\frac{\partial}{\partial z}\left(e^{\frac{z}{b}} - \frac{1}{2}\left(e^{-\frac{(r+z)-l}{b}} - e^{-\frac{u-(r+z)}{b}}\right)e^{\frac{z}{b}}\right) \\
&= \frac{1}{C(r)}\frac{\partial}{\partial z}\left(e^{\frac{z}{b}} - \frac{1}{2}\left(e^{\frac{-r+l}{b}} - e^{\frac{-u+r+2z}{b}}\right)\right) \\
&= \frac{1}{C(r)b}\left(1 - e^{-\frac{u-r-z}{b}}\right)e^{\frac{z}{b}}.
\end{aligned}$$

As $b > 0$, we then see that $\frac{\partial}{\partial z}F(r,z) \geq 0$ when $r+z \leq u$. Then we prove that $\frac{\partial}{\partial r}F(r,z) \leq 0$ when $z \geq 0$. First we note,

$$\frac{\partial}{\partial r}C(r+z) = \frac{1}{2b}\left(e^{-\frac{-r+z-l}{b}} - e^{-\frac{u-r-z}{b}}\right).$$

We find,

$$\begin{aligned}
\frac{\partial}{\partial r}F(r,z) &= \frac{e^{\frac{z}{b}}}{C(r)^2}\left(C(r)\frac{\partial}{\partial r}C(r+z) - C(r+z)\frac{\partial}{\partial r}C(r)\right) \\
&= \frac{e^{\frac{z}{b}}}{2bC(r)^2}\left(e^{-\frac{r-l}{b}}(e^{-\frac{z}{b}} - 1) + e^{-\frac{u-l-z}{b}} + e^{-\frac{u-r}{b}}(1 - e^{\frac{z}{b}})\right. \\
&\left. - e^{-\frac{u-l+z}{b}}\right) \\
&= \frac{e^{\frac{z}{b}}\left(\left(e^{-\frac{z}{b}} - 1\right)\left(e^{\frac{u-r}{b}} - 1\right) + \left(1 - e^{\frac{z}{b}}\right)\left(e^{\frac{r-l}{b}} - 1\right)\right)}{2be^{\frac{u-l}{b}}C(r)^2}.
\end{aligned}$$

Since $r \in [l,u]$, it proves that $e^{\frac{u-r}{b}}, e^{\frac{r-l}{b}} > 1$. When $z \geq 0$, it shows that $e^{-\frac{z}{b}} < 1$ and $e^{\frac{z}{b}} > 1$. Therefore, $\frac{\partial}{\partial r}F(r,z) \leq 0$ when $z \geq 0$.

As $\frac{\partial}{\partial r}F(r,z) \leq 0$, the maximum value of $F(r,z)$ at a fixed z^0 is attained at the smallest possible value of r, i.e $r = l$.

$$\max_{\substack{r, r+z^0 \in [l,u] \\ 0 \leq z^0 \leq \Delta f}} F(r,z^0) = \max_{0 \leq z^0 \leq \Delta f} \frac{C(l+z^0)}{C(l)}e^{\frac{z^0}{b}}.$$

Then, as $\frac{\partial}{\partial z}F(l,z) \geq 0$, the maximum value of $F(l,z)$ is attained at the largest possible z, i.e

$z = \Delta f$,

$$\max_{0 \le z \le \Delta f} \frac{C(l+z)}{C(l)} e^{\frac{z}{b}} = \frac{C(l+\Delta f)}{C(l)} e^{\frac{\Delta f}{b}}.$$

□

We define ΔC for later use:

$$\Delta C = \frac{C(l+\Delta f)}{C(l)}.$$

Theorem 1. *When scale parameter $b \ge \frac{\Delta f}{\varepsilon - log \Delta C}$, it is sufficient to show that the Bounded Laplace mechanism W satisfies ε-local differential privacy*

Proof. Assume that r and r' are a pair of possible inputs to a Bounded Laplace mechanism and $r' = r + z$. Let $0 \le z \le \Delta f$. r^* represents a perturbed output produced by the BLP mechanism. Given the domain of the perturbed output is $[l, u]$, we can note that,

$$Pr(W(r) \in [l,u]) = \frac{1}{C(r)} Pr(M(r) \in [l,u]),$$

where M represents the Laplace mechanism.

We aim to find a condition under which W satisfies ε-local differential privacy. Based on the LDP definition, we can note that,

$$Pr(W(r) \in [l,u]) \le e^{\varepsilon} Pr(W(r') \in [l,u]),$$
$$\frac{1}{C(r)} Pr(M(r) \in [l,u]) \le e^{\varepsilon} \frac{1}{C(r')} Pr(M(r') \in [l,u]).$$

Given that $Pr(M(r) \in [l,u]) = \int_l^u \frac{1}{2b} e^{-\frac{|r^* - r|}{b}} dr^*$, we have,

$$\frac{1}{C(r)} \int_l^u \frac{e^{-\frac{|r^* - r|}{b}}}{2b} dr^* \le e^{\varepsilon} \frac{1}{C(r')} \int_l^u \frac{e^{-\frac{|r^* - r'|}{b}}}{2b} dr^*. \tag{3.1}$$

A lower bound for $e^{\varepsilon}\frac{1}{C(r')}\int_l^u \frac{e^{-\frac{|r^*-r'|}{b}}}{2b}dr^*$ can be obtained using the triangle inequality, i.e.

$$
\begin{aligned}
&| r^* - r' | \leq | r^* - r | + | r' - r |,\\
&e^{\varepsilon}\frac{1}{C(r')}\int_l^u \frac{e^{-\frac{|r^*-r'|}{b}}}{2b}dr^* \geq e^{\varepsilon}\frac{1}{C(r')}\int_l^u \frac{e^{-\frac{|r^*-r|+|r'-r|}{b}}}{2b}dr^*,\\
&e^{\varepsilon}\frac{1}{C(r')}\int_l^u \frac{e^{-\frac{|r^*-r'|}{b}}}{2b}dr^* \geq e^{\varepsilon-\frac{|r'-r|}{b}}\frac{1}{C(r')}\int_l^u \frac{e^{-\frac{|r^*-r|}{b}}}{2b}dr^*.
\end{aligned}
$$

To ensure Eq. (3.1) hold, it is sufficient to show that:

$$
\frac{1}{C(r)}\int_l^u \frac{1}{2b}e^{-\frac{|r^*-r|}{b}}dr^* \leq e^{\varepsilon-\frac{|r'-r|}{b}}\frac{1}{C(r')}\int_l^u \frac{1}{2b}e^{-\frac{|r^*-r|}{b}}dr^*. \tag{3.2}
$$

The inequality given by Eq. (3.2) can be further reduced as,

$$
\frac{C(r)}{C(r')}e^{\varepsilon-\frac{|r'-r|}{b}} \geq 1.
$$

From Lemma 2 we can note that,

$$
\frac{C(r')}{C(r)}e^{\frac{|r'-r|}{b}} \leq \Delta C e^{\frac{\Delta f}{b}},
$$

Equivalently,

$$
\frac{C(r)}{C(r')}e^{\varepsilon-\frac{|r'-r|}{b}} \geq \frac{1}{\Delta C}e^{\varepsilon-\frac{\Delta f}{b}}.
$$

We find a lower bound for $\frac{C(r)}{C(r')}e^{\varepsilon-\frac{|r'-r|}{b}}$ and proceed to find a condition for Eq. (3.2) to hold.

To make Eq. (3.2) hold, it is sufficient to show that,

$$
1 \leq e^{\varepsilon-\frac{\Delta f}{b}}\frac{1}{\Delta C}.
$$

or equivalently,

$$
b \geq \frac{\Delta f}{\varepsilon - \log(\Delta C)}.
$$

□

Theorem 1 provides the scale parameter for BLP to satisfy ε-local differential privacy. It also demonstrates that BLP cannot satisfy ε-local differential privacy when inheriting the scale parameter from the Laplace mechanism. We use BLP as an input rating perturbation mechanism in our recommendation system. The input perturbation mechanism calibrates the magnitude of noise added to original ratings according to the sensitivity given by $\Delta f = u - l$.

We define ΔC as:

$$\begin{aligned}\Delta C &= \frac{C(l + \Delta f)}{C(l)} \\ &= \frac{1 - \frac{1}{2}(e^{-\frac{\Delta f}{b}} + e^{-\frac{u - \Delta f - l}{b}})}{1 - \frac{1}{2}(1 + e^{-\frac{u-l}{b}})}.\end{aligned}$$

When $\Delta f = u - l$,

$$\Delta C = \frac{1 - \frac{1}{2}(1 + e^{-\frac{(u-l)}{b}})}{1 - \frac{1}{2}(1 + e^{-\frac{(u-l)}{b}})} = 1.$$

Thus $\log \Delta C = 0$.

Therefore we can conclude that a sufficient condition needed for the BLP mechanism to satisfy ε-local differential privacy in our recommendation system can be given by:

$$b \geq \frac{u - l}{\varepsilon}.$$

Algorithm 3 details how a perturbed rating is generated using the BLP mechanism.

Algorithm 3 BLP Mechanism for Noise Sampling

1: **Input to the Mechanism: Original Rating** (r)
2: **Output of the Mechanism: Perturbed Rating** (r^*)
3: A noise value is generated from the Laplace distribution with mean 0 and variance of b - $Lap(0, b)$
4: Add noise to original rating to obtain perturbed rating: $r^* = r + Lap(0, b)$
5: **If** ($r^* \in (l, u)$):
6: Perturbed rating is set to r^*
7: **else**
8: repeat Step 3 until ($r^* \in (l, u)$)
9: **Return** Perturbed rating to the TPSP

BLP Noise Distribution

The noise distribution of the BLP mechanism can be theoretically derived for any given dataset of true input ratings. Consider a discrete rating system containing h evenly distributed discrete ranks with step size c. The rank set is denoted by $\mathcal{Q} = \{Q_1, .., Q_{h-1}, Q_h\}$, and $\mid Q_{i+1} - Q_i \mid = c, 1 \leq i \leq h-1$. Let r be a true rating, its corresponding perturbed rating is $r^* = r + n$, where n is the random noise drawn by the BLP mechanism. Since r^* can only take values in the set $\mathcal{Q}$, i.e. $Q_1 \leq r^* \leq Q_h$, we have the noise range for input rating r as $Q_1 - r \leq n \leq Q_h - r$. Define the probability θ_r as:

$$\theta_r = Pr\big(Q_1 - r \leq n < Q_h - r\big),$$

θ_r represents the probability that the noise variable n falls into the interval $(Q_1 - r, Q_h - r)$ given an input rating r. The input rating r takes values in a finite set $\mathcal{Q}$. θ_r can thus be expanded as:

$$\theta_r = \sum_{Q_i \in \mathcal{Q}} Pr(r = Q_i) Pr\big(Q_1 - Q_i \leq n < Q_h - Q_i | r = Q_i\big),$$

where $Pr(r = Q_i)$ is the probability that the input rating equals to Q_i, and $Pr\big(Q_1 - Q_i \leq n < Q_h - Q_i | r = Q_i\big)$ is the conditional probability that the BLP noise lies within the interval $(Q_1 - Q_i \leq n < Q_h - Q_i)$ under the condition that the input rating is Q_i. Note that not all the input ratings in $\mathcal{Q}$ lead to the noise n falling within this particular range. When the perturbed rating $r^* \notin \mathcal{Q}$, the conditional probability $Pr\big(Q_1 - Q_i \leq n < Q_h - Q_i | r = Q_i\big)$ yields 0.

The noise added by the BLP mechanism over all possible input ratings in $\mathcal{Q}$ is a random variable ranging within $(Q_1 - Q_h \leq n < Q_h - Q_1)$. We will then divide the range into equal intervals. The length of each interval is the rank step size of c. The probability that the noise variable lies within each interval is given as follows:

$$\begin{aligned} &Pr\Big(Q_1 - Q_{h-t} \leq n < Q_1 - Q_{h-t-1}\Big) = \sum_{i=h-t}^{h} Pr(r = Q_i) \cdot \\ &Pr\Big(Q_1 - Q_{h-t} \leq n < Q_1 - Q_{h-t-1} | r = Q_i\Big) \\ &\forall t = 0, \cdots, h-2, \end{aligned} \tag{3.3}$$

and

$$
\begin{aligned}
&Pr\Big(Q_h - Q_{t+1} \leq n < Q_h - Q_t\Big) = \sum_{i=1}^{t} Pr(r = Q_i) \cdot \\
&Pr\Big(Q_h - Q_{t+1} \leq n < Q_h - Q_t | r = Q_i\Big) \\
&\forall t = 2, \cdots, h-1,
\end{aligned} \tag{3.4}
$$

The conditional probability is given by

$$
\begin{aligned}
&Pr\Big(n \in [Q_1 - Q_{h-t}, Q_1 - Q_{h-t-1}) | r = Q_i\Big) \\
&= \int_{Q_1 - Q_{h-t}}^{Q_1 - Q_{h-t-1}} \frac{1}{C_r} \frac{1}{2b} e^{-\frac{|r^* - x|}{b}} dr^* \\
&= \frac{1}{C_r} \frac{1}{2b} (e^{Q_1 - Q_{h-t-1})} - e^{Q_1 - Q_{h-t}}) \\
&\text{for} \quad t = 0, \cdots, h-2.
\end{aligned}
$$

and

$$
\begin{aligned}
&Pr\Big(n \in [Q_h - Q_{t+1}, Q_h - Q_t) | r = Q_i\Big) \\
&= \int_{Q_h - Q_{t+1}}^{Q_h - Q_t} \frac{1}{C_r} \frac{1}{2b} e^{-\frac{|r^* - x|}{b}} dr^* \\
&= \frac{1}{C_r} \frac{1}{2b} (e^{Q_h - Q_t} - e^{Q_h - Q_{t+1}}) \\
&\text{for} \quad t = 2, \cdots, h-1.
\end{aligned}
$$

where $C_r = 1 - \frac{1}{2}\Big(e^{-\frac{Q_i - Q_1}{b}} + e^{-\frac{Q_h - Q_i}{b}}\Big)$.

3.1.2 Noise Estimation with Mixture of Gaussians

The MoG model is widely used to approximate probability distributions with no closed-form expression. In image processing this model is used for image segmentation (Vidal et al., 2008), image compression (Turk and Pentland, 1991) and background subtraction (Meng and De La Torre, 2013). We propose an MoG with an MF recommendation model to estimate the noise added to the true ratings and predict missing ratings. Since a multivariate Gaussian distribution can model the uncertainty of a noise data point, MoG is a good solution for noise estimation.

Since we add BLP noise to each true rating in the rating matrix R, the perturbed rating matrix R^* can thus be given by:

$$R^* = R + N.$$

We aim to find a mixture of K Gaussian components which best represent the noise distribution. We assume that each noise data point n_{ij} in N is drawn from a Gaussian distribution $\mathcal{N}(n_{ij} \mid 0, \sigma_k^2)$ where σ_k is the standard deviation of the k-th Gaussian component. The mixture of K Gaussian components representing the noise data point n_{ij} can thus be given by:

$$p(n_{ij} \mid \Pi, \Sigma) \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(n_{ij} \mid 0, \sigma_k^2),$$

in which π_k ($\sum_{k=1}^{K} \pi_k = 1$) is the mixture proportion representing the probability that n_{ij} is drawn from the k-th mixture component. $\Pi = (\pi_1, \pi_2, ...\pi_k)$ and $\Sigma = (\sigma_1, \sigma_2, ...\sigma_k)$. As given in preliminaries, each known rating in original rating matrix R can be approximated using MF as:

$$r_{ij} = (u_i^T)v_j.$$

Hence each rating r_{ij}^* in the perturbed rating matrix can be given by:

$$r_{ij}^* = r_{ij} + n_{ij} = (u_i^T)v_j + n_{ij}.$$

Subsequently, the probability distribution of perturbed rating r_{ij}^* can then be given by:

$$p(r_{ij}^* \mid u_i, v_j, \Pi, \Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}(r_{ij}^* \mid (u_i^T)v_j, \sigma_k^2).$$

The likelihood of R^* can thus be given by:

$$p(R^* \mid V, U, \Sigma, \Pi) = \prod_{i,j \in \Omega} \sum_{k=1}^{K} \pi_k \mathcal{N}(r_{ij}^* \mid (u_i^T)v_j, \sigma_k^2),$$

where Ω represents the set of non-missing data points in perturbed rating matrix R^*. Given the likelihood, next, we derive the maximum likelihood estimates of the model parameters V, U, Σ

and Π for the perturbed rating matrix R^*, i.e.:

$$\begin{aligned} &\max_{V,U,\Sigma,\Pi} \mathcal{L}(R^* \mid V, U, \Sigma, \Pi) \\ &\quad = \sum_{i,j\in\Omega,} \log \sum_{k=1}^{K} \Big(\pi_k \mathcal{N}(r^*_{ij} \mid (u_i^T) v_j, \sigma_k^2) \Big). \end{aligned} \tag{3.5}$$

The log-likelihood can be simplified as :

$$\max_{U,V,\Pi,\Sigma} \sum_{i,j\in\Omega} \sum_{k=1}^{K} \gamma_{ijk} \Big(\log \pi_k - \log \sqrt{2\pi}\sigma_k - \frac{(r^*_{ij} - (u_i^T) v_j)^2}{2\sigma_k^2} \Big). \tag{3.6}$$

3.1.3 Expectation Maximisation for MoG

We use Expectation Maximization (EM) (Dempster et al., 1977) to evaluate and compute model parameters V, U, Σ and Π to maximise the likelihood function given by Eq. (3.5). The EM is an iterative algorithm that can be summarised as follow:

- Initialise the model parameters
- Evaluate the initial value of log-likelihood
- Expectation (E-Step) : Evaluate the posterior responsibilities using the current model parameters
- Maximisation (M-Step) : Re-estimate the model parameters using the current posterior responsibilities

EM algorithm updates the parameters and alternates E-step and M-step until convergence. The standard EM algorithm estimates the mean of each cluster at every iteration. The clusters share the same parameters U and V in our system.

At first, we randomly initialise the model parameters V, U, Σ and Π to estimate the posterior responsibilities of K Gaussian components. In E-step, we estimate the posterior responsibility for each noise point n_{ij} using the current model parameters V, U, Σ and Π as:

$$\gamma_{ijk} = \frac{\pi_k \mathcal{N}(r^*_{ij} \mid (u_i^T) v_j, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(r^*_{ij} \mid (u_i^T) v_j, \sigma_k^2)}. \tag{3.7}$$

The posterior responsibility reflects the probability that k-th Gaussian component produces the

noise point n_{ij}. Then in M-step, we re-estimate each model parameter V, U, Σ and Π based on the posterior responsibilities γ_{ijk} from E-step. We first update Π and Σ:

$$S_k{}^{(x+1)} = \sum_{i,j \in \Omega} \gamma_{ijk}{}^{(x)},$$

$$\pi_k{}^{(x+1)} = \frac{S_k{}^{(x+1)}}{S},$$

$$\sigma_k^2 = \frac{1}{S_k{}^{(x+1)}} \sum_{i,j \in \Omega} \gamma_{ijk}{}^{(x)} (r_{ij}^* - (u_i^T) v_j)^2, \tag{3.8}$$

where S is the total number of non-missing data points, $S_k^{(x+1)}$ is the sum of γ_{ijk} for k-th Gaussian component, and x is the total number of iterations EM algorithm runs until convergence. Then we update the model parameters U and V. We can rewrite the portion in Eq. (3.6) which is related to U and V as:

$$\begin{aligned}
&\max_{V,U} \sum_{i,j \in \Omega} \sum_{k=1}^{K} \gamma_{ijk} \left(- \frac{(r_{ij}^* - (u_i^T) v_j)^2}{2\sigma_k^2} \right) \\
&= - \sum_{i,j \in \Omega} \left(\sum_{k=1}^{K} \frac{\gamma_{ijk}}{2\sigma_k^2} \right) (r_{ij}^* - u_i^T v_j)^2 \\
&= - \sum_{i,j \in \Omega} w_{ij} (r_{ij}^* - u_i^T v_j)^2,
\end{aligned} \tag{3.9}$$

where w_{ij} represents the weight for each true rating r_{ij}, given by:

$$w_{ij} = \begin{cases} \sqrt{\sum_{k=1}^{K} \frac{\gamma_{ijk}}{2\sigma_k^2}}, & \text{if } i, j \in \Omega \\ 0, & \text{if } i, j \notin \Omega. \end{cases}$$

Eq. (3.9) is equivalent to a weighted low-rank MF problem as given below:

$$\min_{U,V} W \odot (X - UV^T)^2.$$

The weighted low-rank MF problems can be solved using methods such as Weighted Low-Rank Approximation (Srebro and Jaakkola, 2003), Damped Newton (Buchanan and Fitzgibbon, 2005)

and Weighted PCA (De La Torre and Black, 2003). We use Weighted PCA in this work to re-estimate model parameters U and V. The EM algorithm stops alternating between E-step and M-step when two consecutive user latent factor matrices U cause a change smaller than the given threshold value or the number of iterations reaches the predefined threshold. Algorithm 4 details how MoG with the MF model estimates noise and predicts missing ratings.

Algorithm 4 Noise Estimation and Rating Prediction Model

1‘

1: **Input: Perturbed Ratings (R^*)**
2: **Output:** U **and** V
3: *Initialisation*: Model parameters U, V, Π and Σ are randomly initialised
4: In E-step posterior responsibility ${\gamma_{ijk}}^{(x)}$ is estimated using Eq. (3.7)
5: **For** Until convergence
6: (M-Step for updating $\Sigma^{(x+1)}$ and $\Pi^{(x+1)}$) Model parameters Σ and Π are computed using Eq. (3.8)
7: (M-Step for estimating V and U) Model parameters U and V are updated using Eq. (3.9)
8: (E-step for posterior responsibility γ_{ijk}) posterior responsibility γ_{ijk} is computed using current model parameters
9: **Return** User and Item latent factor matrices U and V

3.2 Evaluation

In this section, we evaluate the effectiveness of our proposed recommendation model through real-world datasets.

3.2.1 Datasets

We use three datasets: Movielens (Harper and Konstan, 2015), Libimseti (Brozovsky and Petricek, 2007) and Jester (Goldberg et al., 2001) in the evaluation. Table 3 provides a detailed view of the datasets. For privacy budget ε, we consider the value range from 0.1 to 3, lower values of privacy budget ε guarantee stronger privacy protection for users.

Table 3: Datasets for BLP-MoG-MF Evaluation

Dataset	Total Ratings	No of Items	No of Users	Rating Scale
Movielens	100k	1682	943	0.5 to 5
Jester	2 Million	100	73,421	-10 to 10
Libimseti	17,359,346	168,791	135,359	1 to 10

3.2.2 Evaluation Metrics

A standard metric used to evaluate the performance of a recommendation system is Root Mean Squared Error (RMSE), which calculates the average error between actual ratings and predicted ratings. RMSE can be estimated as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n-1} (r_i - \hat{r_i})^2}{n}},$$

where r_i is the actual rating, $\hat{r_i}$ is the predicted rating and n is the total number of ratings in the aggregated dataset.

However, the performance of a recommendation system relies heavily on the utility - i.e. the suitability between the recommended items and the user's expectations. That means the top-N recommendations made by the recommendation system should be as close as possible to the user's preferences. Hence, RMSE cannot be used to evaluate the performance of top-N recommendations. Therefore, utility metrics such as recall and precision can be used as alternative methodologies to assess the top-N recommendation performance. As shown in Table 4, a confusion matrix is created to obtain these utility metrics. The confusion matrix relies on the following values:

- True Positive: The item is recommended to the user and is preferred by the user.
- True Negative: The item is not recommended to the user and is not preferred by the user.
- False Positive: The item is recommended to the user but not preferred by the user.
- False Negative: The item is not recommended to the user but is preferred by the user.

Table 4: Confusion Metric.

Items	Preferred by Users	Not Preferred by Users
Recommended	True Positive (TP)	False Positive (FP)
Not Recommended	False Negative (FN)	True Negative (TN)

Precision identifies the proportion of items that are correctly recommended to the users out of all recommended items. Precision. Precision can be computed as follows:

$$\text{Precision} = \frac{\#\text{True Positives}}{\#\text{True Positives} \ + \ \#\text{False Positives}},$$

Recall identifies the proportion of items that are correctly recommended to the users out of all the preferred items. Recall can be computed as follows:

$$\text{Recall} = \frac{\#\text{True Positives}}{\#\text{True Positives} \ + \ \#\text{False Positives}}.$$

F1 score is a metric that gives the weighted average of recall and precision and can be computed as :

$$\text{F1-score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}.$$

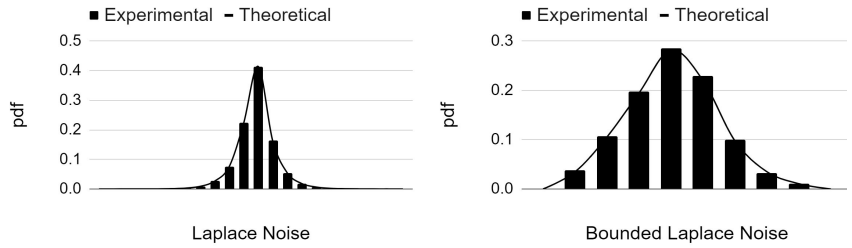
3.2.3 Results

Noise Distribution Evaluation

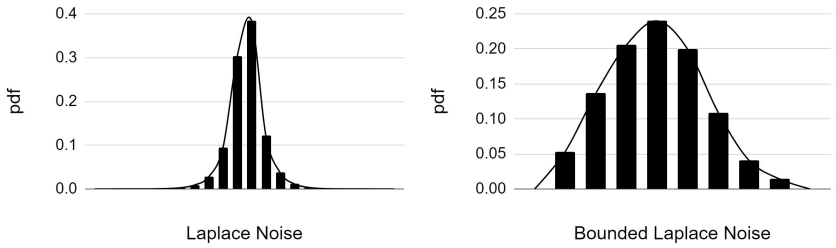
We derive the BLP noise distribution theoretically in section 3.1.1. This section shows that the noise distribution of Laplace and BLP mechanisms are distinct. We generate 100,000 random noise samples using BLP and Laplace mechanisms for the Movielens dataset while positioning their privacy budget ε to 0.1 and 1. Figure‘. 6a and 6b display the probability of noise samples drawn by Laplace and Bounded Laplace mechanisms. From the probability density functions, we note that the noise distribution of the two mechanisms is distinct. We also plot the BLP noise distribution curve based on our derived noise distribution expressions given by Eq. (3.3) and (3.4). Fig. 6a and 6b show that the theoretical derivation of distribution follows the experimental distributions exactly.

Influence of BLP on Predictive Accuracy

In this experiment, we demonstrate that using BLP as an input perturbation mechanism does play a significant role in obtaining higher predictive accuracy. We measure the RMSE when either BLP or Laplace act as the input perturbation mechanism while using the same rating prediction model (MoG or SVD). Fig. 7a and 7b display the resulting RMSE metric values for Movielens and Jester datasets respectively. The BLP mechanism results in higher recommendation accuracy than the Laplace mechanism for both rating prediction models.



(a) $\varepsilon = 1$



(b) $\varepsilon = 0.1$

Figure 6: Laplace vs Bounded Laplace Noise Distribution

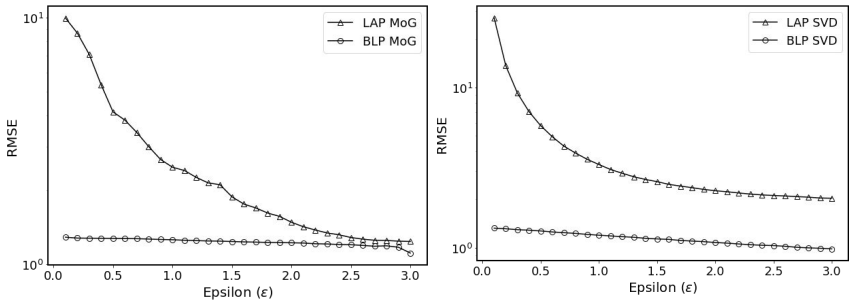
Influence of MoG on Predictive Accuracy

In this experiment, we demonstrate that employing MoG in our recommendation model aids in improving predictive accuracy for lower values of privacy budget ε. We measure the RMSE values when using the MoG or the SVD for rating prediction while using the same data perturbation mechanism. Fig. 8a and 8b display the resulting RMSE values for Movielens and Jester datasets respectively. For both datasets, the predictive accuracy from the MoG prediction model is much higher than SVD.

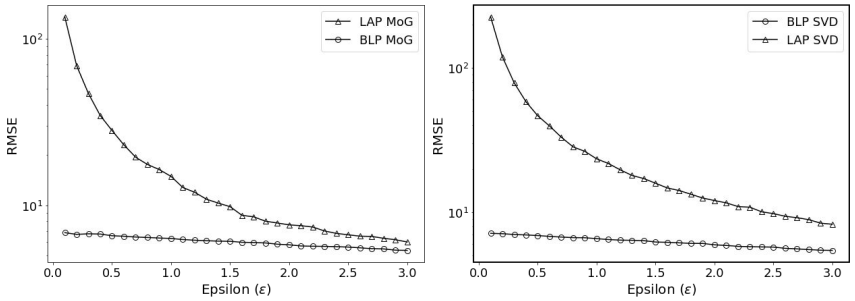
Predictive Accuracy Comparison

We compare the predictive accuracy of our recommendation model with other existing local differentially private recommendation models such as:

- Input Perturbation Method (ISGD) (Berlioz et al., 2015): This method perturbs the user's original ratings locally using the Laplace mechanism. However, they apply a truncation method to ensure that the perturbed rating falls within a predefined domain. The noised ratings that fall out of a predefined range are clamped to either the lower or upper bound of the rating domain using a threshold value. ISGD method uses MF for rating prediction at the TPSP's side.



(a) Movielens

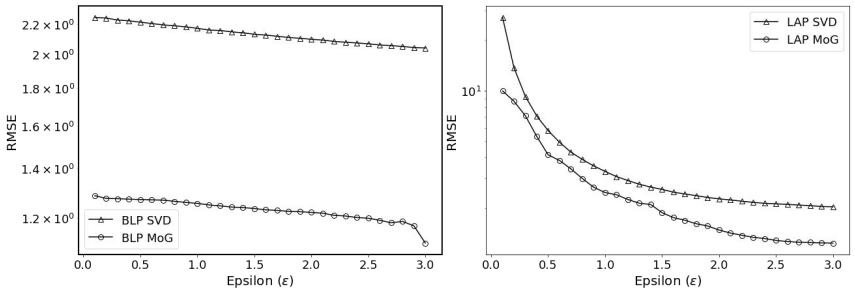


(b) Jester

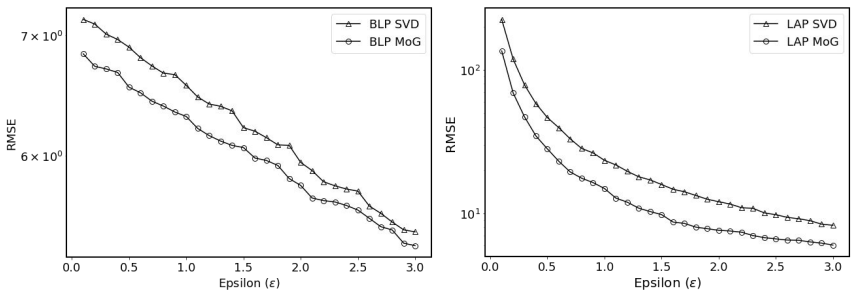
Figure 7: Bounded Laplace Mechanism vs Laplace Mechanism RMSE Comparison

- Private Gradient-Matrix Factorization (PG-MF) (Shin et al., 2018): This approach uses MF to perform recommendations. In this approach, the user computes user latent factors locally without submitting them to the TPSP. The TPSP estimates the item latent factors after collecting gradients from the users. On the other hand, users compute a perturbed gradient and submit that to the TPSP. The TPSP aggregates the perturbed gradient from all the users and then updates the item latent factors accordingly.

We use Non-Private MF as the baseline method as it does not use any local perturbation mechanism to perturb the user's original ratings. Instead, the MF algorithm uses actual ratings to predict missing ratings. The baseline method provides a lower bound RMSE value for predictive error. Our recommendation model (BLP-MoG-MF) uses BLP as the input perturbation mechanism and MoG-MF as the recommendation algorithm. The BLP-MoG-MF method uses the objective function specified by Eq. (2.3) to obtain latent factor matrices. ISGD and PG-MF methods also perform rating predictions using the same objective function. To maintain the fairness of comparison, we did not compare our results with recommendation models that use different approaches to predict missing ratings. Firstly, we compare BLP-MoG-MF with PG-MF. We vary the privacy budget ε from 0.1 to 1.6 for the Movielens dataset. Fig. 9 displays the RMSE values for BLP-MoG-MF,



(a) Movielens



(b) Jester

Figure 8: MoG vs SVD Prediction Model RMSE Comparison

PG-MF and the baseline method. As expected, when the privacy budget increases, predictive accuracy for all privacy protection methods increases. Because when privacy budget ε increases, the magnitude of privacy loss LDP mechanism permits increases, which causes a rise in the predictive accuracy. More importantly, BLP-MoG-MF provides a lower RMSE than PG-MF for the same privacy budget ε.

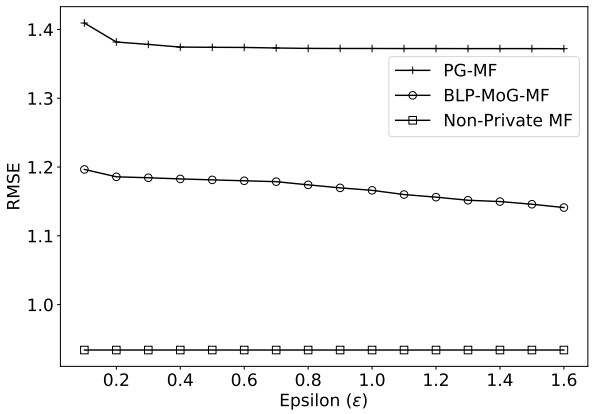


Figure 9: PG-MF vs BLP-MoG-MF RMSE Comparison for Movielens

Then, we compare BLP-MoG-MF with the ISGD method for Movielens, Libimseti and Jester datasets. We vary the privacy budget ε from 0.1 to 3 for all the datasets in this simulation. Fig.

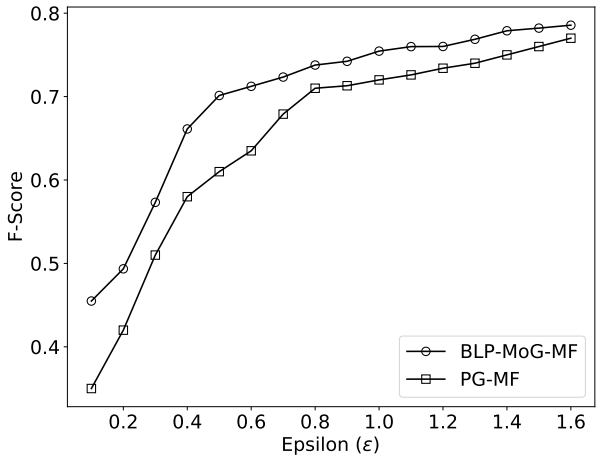


Figure 10: PG-MF vs BLP-MoG-MF F1-Score Comparison for Movielens

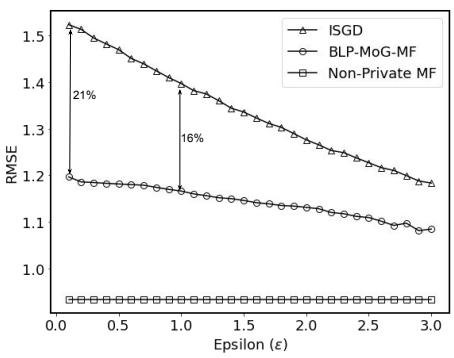
11a, 11b and 11c display the RMSE values for BLP-MoG-MF, ISGD and the baseline methods. The results show that BLP-MoG-MF outperforms ISGD significantly for all values of the privacy budget ε. This trend implies that our method guarantees higher data utility for all the values of privacy budget ε.

Privacy-Utility Trade-off Analysis

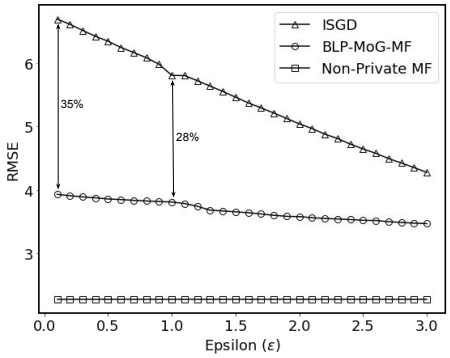
We use F1-Score as a utility metric and the privacy budget ε as the privacy loss metric to evaluate our system. We compute the F1-Score by comparing the top 10 items that our LDP-based recommendation system recommends against the top 10 items recommended by other algorithms. Fig. 10 demonstrates the F1-Score values for BLP-MoG-MF and PG-MF methods. Similar to RMSE values, the F1-Score value increases as the privacy budget ε increases. The F1-Score results also show that BLP-MoG-MF provides more accurate recommendations than PG-MF for all privacy budget values ε. Likewise, Fig. 12a, 12b and 12c illustrate the F1-Score values for Movielens, Jester and LibimSeti datasets for BLP-MoG-MF and ISGD methods. There is a substantial increase in F1-Score as the privacy budget ε increases for all three datasets and both methods. Again, for all three datasets, the F1-Score of the BLP-MoG-MF is higher than the ISGD.

Analysis of Communications Cost

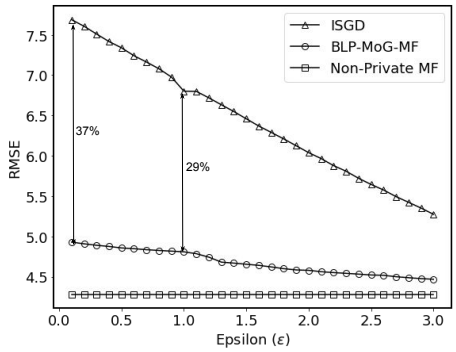
We compare the communication cost incurred in our approach to recommendation models proposed by (Shin et al., 2018) and (Berlioz et al., 2015). Table 5 summarises the analysis. Both



(a) Movielens

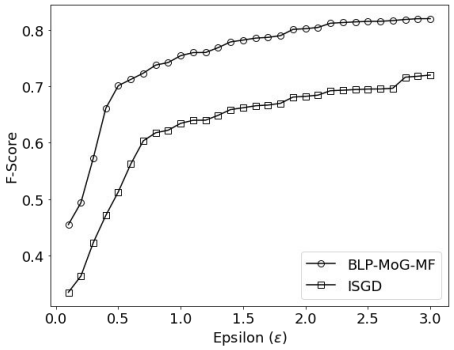


(b) Jester

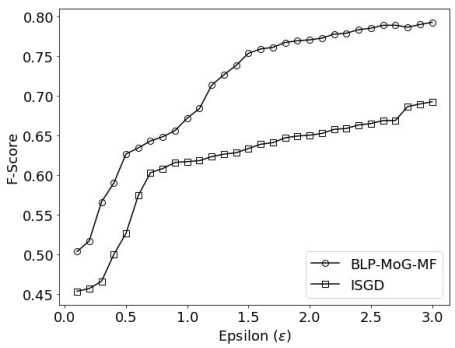


(c) LibimSeti

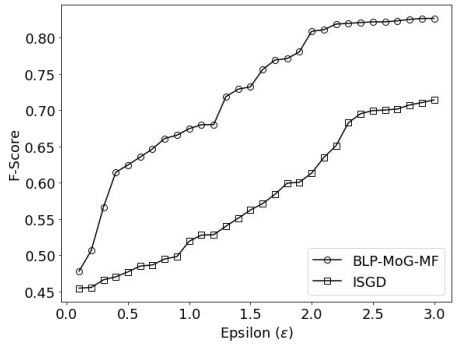
Figure 11: BLP-MoG-MF vs ISGD RMSE Comparison



(a) Movielens



(b) Jester



(c) LibimSeti

Figure 12: BLP-MoG-MF vs ISGD RMSE Comparison

Table 5: Comparison of Communication Cost for Movielens

Recommendation Model	User to TPSP	TPSP to User
BLP-MoG-MF	1 bit	No transfer
ISGD	1 bit	No transfer
PG-MF	1 bit	0.15MB

BLP-MoG-MF and ISGD methods require the user to transfer a perturbed rating whenever the user rates an item. In the PG-MF method, the user transmits the perturbed stochastic gradient to the TPSP over multiple data transmission iterations. Both BLP-MoG-MF and ISGD methods do not require the TPSP to transmit any data back to the user. However, the TPSP transmits an updated item latent factor matrix back to the user at each iteration in the PG-MF approach. This exchange between the TPSP and the user continues until the number of iterations reaches a pre-defined threshold value. We assume a single rating is 1 bit. The estimated size of the transmitted data for each iteration for the PG-MF method is approximately 0.15 MB for the Movielens dataset (Shin et al., 2018). The comparison shows that we significantly reduce the communication cost in our proposed model compared to other local differential private recommendation models.

3.3 Conclusion

In our work, we have proposed a recommendation model under the consideration of an untrustworthy TPSP. We have used BLP as a local input perturbation mechanism and MoG-MF for noise estimation and rating prediction. Our proposed recommendation model can improve predictive accuracy and guarantees strong user privacy compared to existing solutions. Besides, our method does not incur any further communication cost to the user side as it only requires the user to transmit the perturbed rating to the TPSP.

Chapter 4

LDP-based Hybrid Recommendation Model

Many works have proposed integrating sentiment analysis with Collaborative Filtering (CF) algorithms to improve the accuracy of recommendation systems. As a result, a Third-Party Service Provider (TPSP) collects reviews and ratings, which is increasingly causing privacy concerns among users. Several works have used Local Differential Privacy (LDP) based input perturbation mechanism to address privacy concerns related to the aggregation of ratings. However, researchers have failed to address whether perturbing just ratings can protect the privacy of users when both reviews and ratings are collected. We answer this question in this chapter by applying an LDP-based perturbation mechanism in a recommendation system that integrates CF with a sentiment analysis model.

We perturb the user's original ratings locally using a Bounded Laplace (BLP) input perturbation mechanism before sending them to the TPSP. A deep learning-based sentiment analysis model is used to analyze user reviews. However, we propose that user reviews be tokenized locally and then sent to the TPSP for aggregation and classification purposes. Since the TPSP only aggregates the tokenized reviews, they cannot infer sensitive information about users without access to their original review texts.

Additionally, the perturbed ratings are used as the input sentiment labels, preventing the TPSP from learning a user's actual sentiment using their tokenized review data. We use Matrix Factor-

ization (MF) with Mixture of Gaussian (MoG) as our CF algorithm. The MF with MoG model that runs at the TPSP estimates the noise added to the aggregated perturbed ratings and simultaneously predicts missing ratings. The results of the empirical study show that our proposed recommendation model significantly improves recommendation accuracy under a strong privacy guarantee.

4.1 Hybrid Recommendation System with LDP

This section describes our proposed LDP-based recommendation system that combines a CF algorithm with a sentiment analysis model and uses BLP as the input perturbation mechanism. The aim is to improve the recommendation accuracy while providing privacy protection to the users. Fig. 13 illustrates the training architecture of the proposed recommendation system. We assume that users can report their actual ratings and reviews anonymously so that the TPSP can display these anonymous reviews and ratings on their platform. We do not discuss the architecture required for anonymous reporting as it is beyond the scope of this thesis. We use the Matrix factorization with a Mixture of Gaussian model (MF-MoG) model as the CF algorithm and Convolutional Neural Networks (CNN) classification model as the multi-class sentiment classification model. The MF with MoG model uses perturbed ratings as the corresponding training data where as multi-class sentiment classification model uses feature vectors extracted from the user reviews and perturbed ratings as the two features of the training data. The predicted rating combination module uses the outputs of both of these models to produce the final predicted rating. The recommendation module uses these final predicted rating to produce recommendations for the corresponding user.

4.1.1 Input Rating Perturbation at User-side

First, users use BLP as an LDP perturbation mechanism to perturb their original ratings. The perturbed ratings are then sent to the TPSP for aggregation. In section 3.1.1 we describe how a perturbed rating is generated using the BLP mechanism. We also show that a sufficient condition for BLP to satisfy ε-local differential privacy in recommendation systems that uses ratings is when the local sensitivity is $\Delta f = l - u$ where l is the minimum and u is the maximum rating in a given rating scale. The BLP mechanism ensures that the perturbed output rating is limited to the rating domain $[l, u]$ and still guarantees that the adversary cannot infer any information about the original

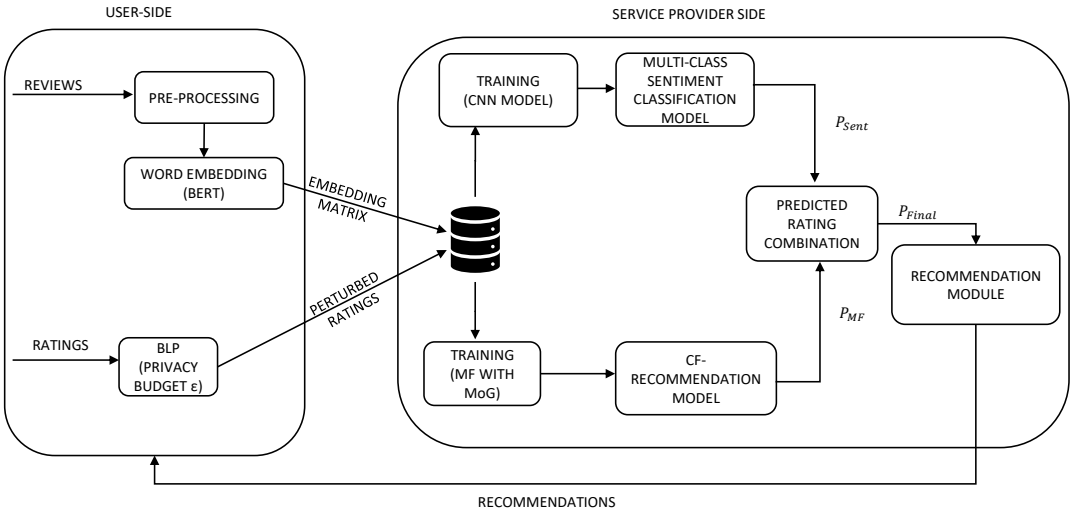


Figure 13: Proposed LDP-based Hybrid Recommendation System Training Architecture

rating by observing the perturbed rating.

4.1.2 Review Pre-processing at User-side

Sentiment analysis models require the input review data to be cleaned and processed before using them in a classification model. The original reviews of users are pre-processed and converted to a tokenized review locally at the user-side before being sent to the TPSP. First, words that lack relevant information, leading and trailing spaces, numbers, punctuation and stop words in the review text are removed. Additionally, the text is converted to lowercase. Then the cleaned review is split into individual words and then lemmatized. The lemmatization process converts a word's inflectional and derivational forms to its common base form. For example, the terms run, runs, and running are converted to the base word run.

4.1.3 Review Tokenizing using Bidirectional Encoder Representations from Transformers

After lemmatization, we use Bidirectional Encoder Representations from Transformers (BERT) to create word embedding in our sentiment analysis model. BERT is a natural language processing model introduced by the Google AI team in 2018 (Devlin et al., 2018). BERT provides a contextualised representation, unlike other word-embedding models such as Word2Vec and GloVe, that generates a single representation for each word in a given input text. BERT maps each word into

a vector of numerical values so that words with similar meanings have a similar representation. BERT uses transformer encoder layers to learn these contextual relations between words in a given text. We set a fixed length S as the maximum length for all the pre-processed review sentences. Each word in a pre-processed review is converted to a word vector. BERT will map each word in a review to a vector which results an $S \times E$ matrix where E is represents the size of the word embedding space. The dimension of the matrix in our model is 768×5. Each user does the review pre-processing locally and sends only the numerical embedding vector matrix to the TPSP. Therefore the actual review text is never revealed to the TPSP.

However, these converted word embeddings can potentially leak the actual sentiment of the user, and in some cases, can be inverted to recover the original user review itself. This potentially harms the privacy guarantee we provide to the users. One approach to tackle this issue would be to use an LDP-mechanism to perturb the word embeddings vector and then transfer them to the TPSP. However, this approach would further affect the accuracy of the proposed system and the privacy budget ε will be used to manually control the noise. Another approach that can be used to tackle this issue is to use Homomorphic Encryption which is a cryptographic primitive that can be used to perform computations over encrypted data without any decryption of the encrypted data.

4.1.4 Perturbed Rating Classification at Server-side

The training dataset used to train the Multi Class Classification model includes tokenized reviews and labels. We used the corresponding rating as the label for each tokenized review. However, due to input rating perturbation, we get continuous values as perturbed ratings. Hence, we cannot use the perturbed ratings as the label. Therefore, on the server side, we classify these ratings into five distinct groups and use those groups as labels for our training dataset. This classification is done as shown in Table 6 and assigns labels accordingly. Each sentiment class is associated with one of the labels from 1 to 5 to be consistent with the rating scale.

4.1.5 Multi Class Classification Using CNN

In our work, we use BERT only as an encoder and CNN model as the decoder to conduct sentiment classifications. Even though BERT can perform sentiment classification, the multi-label classification layer must be retrained on top of the transformer to perform sentiment prediction.

Table 6: Perturbed Rating Classification

Perturbed Rating	Sentiment Class	Label
1-1.49999	Negative	1
1.5-2.49999	Somewhat Negative	2
2.5-3.49999	Neutral	3
3.5-4.49999	Somewhat Positive	4
4.5 - 5	Positive	5

Fig. 14 illustrate the hybrid sentiment analysis model used in our recommendation system.

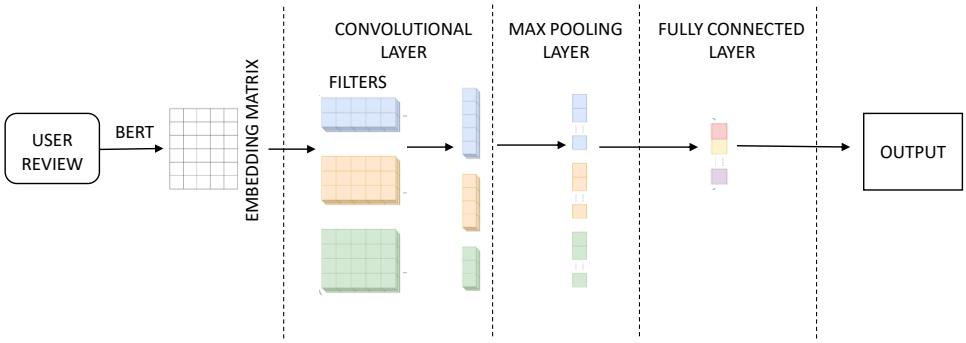


Figure 14: Hybrid Sentiment Analysis

We use CNN model for multi-class classification purposes at the TPSP's end as it is effective in text review classification (Salinca, 2017). Since the CNN model uses classified perturbed ratings as the sentiment labels, we hide the true sentiment of the users from the TPSP. This step adds another layer of privacy to the sentiment analysis model and offers plausible deniability to users. However, using classified perturbed ratings as the sentiment labels will affect the accuracy of the CNN classification model as the sentiments do not reflect a user's preferences. The utility-privacy trade-off in this case is controlled by the privacy-budget ε. For the lower values of the privacy-budget ε, the CNN classification model will yield lower classification accuracy as more noise is added to the actual ratings. However, the CNN classification model will yield higher classification accuracy for higher values of the privacy-budget ε as the noise added to actual ratings is comparatively small. Smaller noise will indicate that the perturbed ratings in some cases might indeed reflect a user's actual preferences.

CNN model learns spatial hierarchical features using three layers that is convolution, pooling and fully connected. The first two layers do feature extraction, and the final fully connected layer maps

the extracted features into a relevant sentiment. The output of the CNN model is the predicted rating P_{Sent}. In our work we used the CNN model architecture proposed by (Pan et al., 2021) for the text classification purposes.

Convolution Layer

The feature vector matrix acts as the input to the convolution layer which extracts features from this matrix to obtain the feature map. In our model we use $2D$ convolution, 64 filters with dimension of 2×5, 64 filters with dimension of 3×5, 64 filters with dimension of 4×5, 64 filters with dimension of 5×5 and the step size of 1. The convolution layer outputs 256 feature vectors.

Pooling Layer

We use max pool to pool each feature vector in to a value. A maximum value from each feature vector is extracted and this maximum value is considered to represent the most important feature. The output of the pooling layer contains the final feature vector.

Fully Connected Layer

The fully connected layer uses the softmax function to classify each review into corresponding classes.

4.1.6 Matrix Factorization with Mixture of Gaussian Model

We use MF-MoG as the CF algorithm to make rating predictions on the TPSP's side. The MF model yields low recommendation accuracy when using perturbed ratings as the input. Hence, we use MoG at the TPSP's side to estimate the noise added to the original ratings to enhance prediction accuracy. Since the post-processing property of LDP states that any further processing of a perturbed output of a differentially private mechanism does not violate the differential privacy principles (Dwork, 2008), the MF-MoG model still can provide privacy protection to users. The output of the MF-MoG model would be the predicted rating P_{MF}.

4.1.7 Predicted Ratings Combination

The predicted ratings combination module combines outputs from the CNN classification model and the MF-MoG model to produce the final predicted rating. The aim is to ensure that we can obtain an accurately predicted rating using both reviews and ratings of the users. The final rating of a user an item is given as:

$$P_{Final} = \beta * P_{MF} + (1 - \beta) * P_{Sent}. \tag{4.1}$$

where P_{Final} is the final predicted rating, P_{MF} is the rating predicted using the MF-MoG model, P_{Sent} is the rating predicted using sentiment analysis and β is the parameter that is used to adjust the importance of each component. The ideal value for β is determined based on empirical analysis and will vary between different datasets.

4.2 Evaluation

In this section, we present the evaluation results of our proposed LDP-based hybrid recommendation model.

4.2.1 Datasets

We use two ratings and review datasets, Amazon Toys and Games and Amazon Instant Video (Ni et al., 2019) to validate the efficiency of the proposed system. Table 7 provides a detailed view of the datasets we used in our evaluation.

Table 7: Datasets for BERT-MF-MoG Evaluation

Dataset	Total Ratings	Total Reviews	Rating Scale
Amazon Toys and Games	2,252,771	167,597	1 to 5
Amazon Instant Video	583,933	37,126	1 to 5

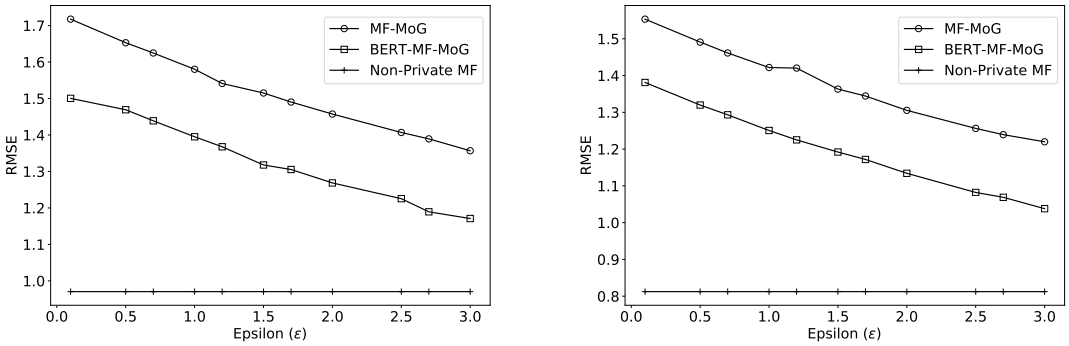
4.2.2 Evaluation Metrics

The privacy budget ε acts as a metric for privacy loss. The lower the privacy budget ε, the higher the privacy guarantee. We consider the value range from 0.1 to 3 for the privacy budget ε. However, if an algorithm uses lower values of privacy budget ε, it decreases the recommendation accuracy. Hence, it plays an integral part in evaluating the trade-off between privacy and utility. The parameter β in Eq. (4.1) controls the role MF-MoG, and the Sentiment analysis model plays in determining the final predicted rating. The lower value of β indicates that the final predicted rating is more reliant on the sentiment analysis model, and the higher value means it relies on the MF-MoG model. We evaluate the trade-off between β and utility by considering the value range from 0.1 to 1 for the parameter β. The utility of a recommendation system is evaluated by how well it predicts the relevance of an item for a user. We use RMSE to measure the predictive accuracy of our system and F-score as the utility metric to measure how well the recommendation systems make recommendations that adapt to a user's choices.

4.2.3 Results

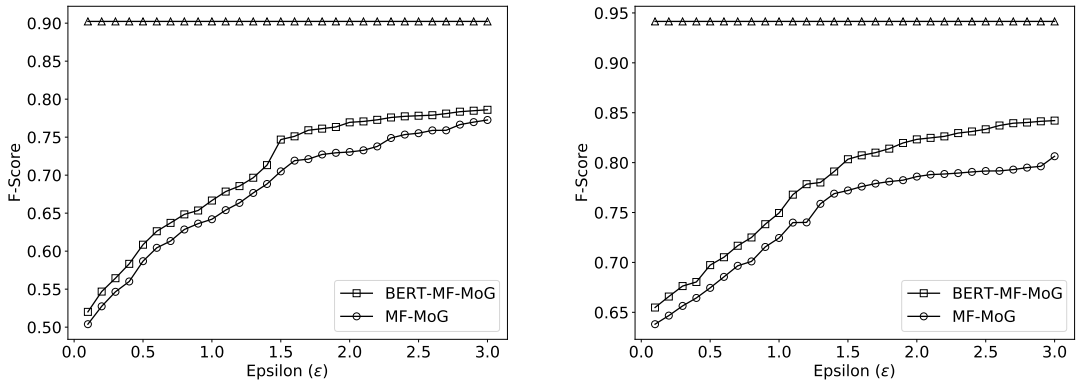
In this experiment, we compare the prediction accuracy of the LDP-based hybrid recommendation model BERT-MF-MoG with the MF-MoG recommendation system proposed in the previous chapter. The privacy budget ε varies from 0.1 to 3, and we use 0.7 as β values for both datasets. Figure. 15a and 15b shows the RMSE values for BERT-MF-MoG ($\beta = 0.7$), MF-MoG and the baseline method (Non-Private MF). The baseline method uses BERT as the sentiment analysis model and MF as the CF algorithm. The baseline method does not perturb the user's original ratings, and the review pre-processing takes place on the service provider's side. As expected, the prediction accuracy of the two privacy-preserving methods, BERT-MF-MoG and MF-MoG, improves when the privacy budget ε increases as an increase in the privacy budget indicates a decrease in noise added to perturb a user's rating. The results also show that the BERT-MF-MoG model increases recommendation accuracy for both datasets for all the values of ε than MF-MoG. This increase is because BERT-MF-MoG combines the sentiment classification with the MF-MoG model, which significantly increases recommendation accuracy.

Figure. 16a and 16b shows the F-score values for BERT-MF-MoG and MF-MoG models. The F-score value increases when the privacy budget ε increases for both models. These F-score values



(a) Amazon Instant Video (b) Amazon Toys and Games

Figure 15: BERT-MF-MoG vs MF-MoG RMSE Comparison



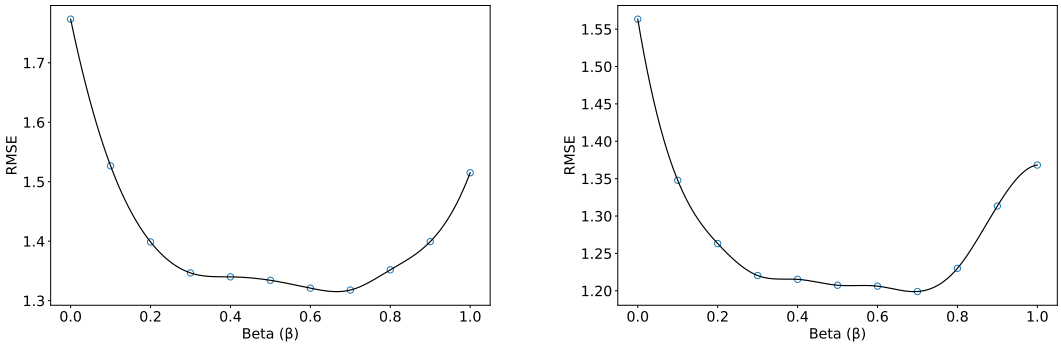
(a) Amazon Instant Video (b) Amazon Toys and Games

Figure 16: BERT-MF-MoG vs MF-MoG F-Score Comparison

demonstrate that the BERT-MF-MoG model provides more accurate recommendations than MF-MoG for all privacy budget values ε.

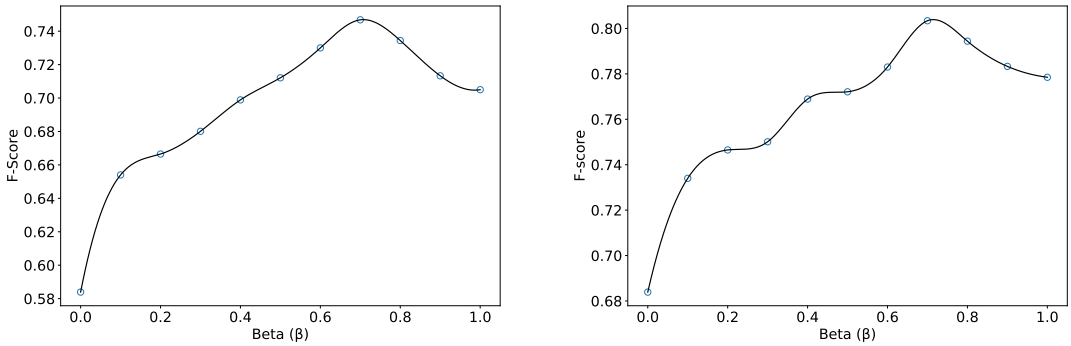
Figure. 17 and Fig. 18 shows the RMSE and the F-score values respectively when varying β while keeping privacy budget at $\varepsilon = 1.5$. These figures show that optimal RMSE and F-score values are obtained when $\beta = 0.7$. When $\beta = 0$, only the output of the sentiment analysis model, P_{Sent}, is used to determine the final predicted rating as indicated in Eq. (4.1), and the output of the MF-MoG model is not used. When β increases from 0, the contribution of the MF-MoG model, which uses a large proportion of ratings, starts to be included in the final predicted rating. Hence, a decrease in RMSE (increase in F-Score) can be observed as β increases in Fig. 17 and 18.

When β increases from 0.7 to 1, the RMSE value increases. This increase is because the predic-



(a) Amazon Instant Video (b) Amazon Toys and Games

Figure 17: β-RMSE Comparison



(a) Amazon Instant Video (b) Amazon Toys and Games

Figure 18: β-F-score Comparison

tion relies more heavily on the MF-MoG model than it should in this range. The significance of sentiment analysis is underestimated. When $\beta = 1$, Eq. (4.1) uses the P_{MF} as final predicted rating. The output of the sentiment classification model P_{Sent} does not play any role in the prediction. The same trend can be observed in Fig. 18 for the F-score value. The larger the β is, the more contribution the MF-MoG model makes, and the more accurate the prediction is until β reaches 0.7, where the highest F-score and lowest RMSE value are obtained. The F-score and RMSE results show that the BERT-MF-MoG model provides the most accurate recommendations when $\beta = 0.7$ for the two datasets.

4.3 Conclusion

This chapter proposes an LDP-based hybrid recommendation system incorporating a deep-learning sentiment analysis model into a CF algorithm. Our proposed system protects users' privacy from a TPSP that aggregates both ratings and reviews and, at the same time, offers substantial utility to the TPSP. The recommendation accuracy of our proposed model has improved by taking advantage of sentiment analysis performed on user reviews. The experiments conducted with two amazon review and rating datasets demonstrated that the utility of our proposed hybrid recommendation system outperforms that of the recommendation system based just on ratings for all the values of privacy budget ε. In future work, we plan to explore using other techniques such as LSTM (Long Short Term Memory Networks) in combination with CNN for sentiment classification to improve recommendation accuracy further.

Chapter 5

Federated Matrix Factorization with Local Differential Privacy

Traditionally Collaborative Filtering (CF) based recommendation systems use explicit feedback such as ratings to predict users' preferences. However, due to the lack of ratings given by users on items, data sparseness has become an issue that affects the utility of these recommendation systems. Hence, more and more Third-Party Service Providers (TPSPs) tend to collect implicit feedback from users to tackle the data sparseness problem. They can easily collect implicit feedback from a user by monitoring their actions such as purchase history, navigation history, browsing habits, etc. Therefore, CF-based recommendation systems no longer depend on the user's explicit feedback data to produce personalised recommendations. However, like explicit feedback, even implicit feedback aggregated from users can cause privacy violations and reveal sensitive information about them to untrustworthy TPSPs.

Federated Learning-based recommendation systems are introduced to tackle these privacy concerns and offer strong privacy protection to users from untrustworthy TPSPs. A federated learning-based recommendation system distributes the model training process among its users, ensuring that users' data never leaves their devices and enhancing their privacy protection. Federated matrix factorization is one of the most frequently used federated CF models to produce personalised recommendations. In this model, the TPSP iteratively updates the item latent factor using the stochastic gradients collected from users. Even though users' implicit or explicit feedback never

leaves their devices, some works have identified that sending model parameters such as stochastic gradient back to the TPSP in federated learning models still reveals sensitive information about the users (Zhu et al., 2019; Geiping et al., 2020; Wei et al., 2020).

Compared to explicit feedback-based federated learning models, only a few works have addressed the intersection of federated learning-based recommendation systems for implicit feedback and privacy. These concerns inspire the requirement for a privacy-first solution for a federated matrix factorization model that uses implicit feedback. We concentrate on two main factors to provide a solution to this problem.

- The TPSP should not be able to aggregate any implicit feedback from the users and, at the same time, should be able to obtain substantial utility from the recommendation system.
- The shared model parameters such as stochastic gradient in federated matrix factorization should be protected.

Additionally, several works have also analysed the factors that affect the convergence of federated learning models while taking into account population size, communication cost and computational cost (Huo et al., 2020; Pathak and Wainwright, 2020). Users are often selected randomly to participate in the training process, and most of the work assumes only a proportion of users take part in training due to limited resource constraints and privacy concerns. Even though it is beyond the scope of this thesis to analyse conditions based on which users are selected, it is necessary to understand the trade-off between population size and utility.

This chapter proposes a Local Differential Privacy (LDP) based federated learning recommendation system for implicit feedback. Our proposed system guarantees strong user privacy protection by incorporating a federated learning approach with LDP. In summary, we have made the following contributions:

- We propose LDP-based federated matrix factorization for implicit feedback. We use BLP as an iterative input perturbation mechanism that perturbs the stochastic gradient sent to the TPSP and provide a sufficient condition for BLP to satisfy ε-local differential privacy in such systems.
- We empirically evaluate the role of the privacy budget ε, the number of iterations k and

the size of user/item set plays in enhancing predictive accuracy using Movielens and Jester datasets. We compare our proposed system with a non-private approach to demonstrate that we can provide substantial utility while providing a stronger privacy guarantee to the users.

5.1 Matrix Factorization for Implicit Feedback

A Matrix Factorization (MF)-based recommendation system is most commonly used to infer users' preferences from explicit feedback. They obtain user and item latent factor matrices from a rating matrix where each element represents a rating given by a user to an item. This recommendation system obtains user and item latent factor matrices by minimising the following loss function:

$$\min_{X,Y} \sum_{i,j} (r_{ij} - x_i^T y_j)^2 + \lambda (\sum_i ||x_i||^2 + \sum_j ||y_j||^2), \tag{5.1}$$

where r_{ij} is the actual rating given by user i for the item j and λ is the regularisation parameter. In Eq. (5.1) x_i represents the relationship between user i and the latent factors in the user latent matrix X. Similarly, y_j represents the relationship between item j and the latent factors in the item latent matrix Y.

However, the loss function given by Eq. (5.1) has to be modified as certain characteristics of implicit data prevent researchers from using the same loss function as explicit feedback in a MF-based recommendation system for various reasons. Assume the implicit feedback p_{ij} of user i on item j is given as:

$$p_{ij} = \begin{cases} 1, & \text{if user interacted with the item,} \\ 0, & \text{if otherwise.} \end{cases}$$

First implicit feedback data does not contain negative feedback. The implicit feedback p_{ij} is defined as either 0 if the user has not interacted with the item or 1 if they have interacted with the item. From this interpretation, we can infer that the user i interacted with item j because they probably liked it. However, if implicit feedback is 0, it does not mean that the user i did not like the item j. When the TPSP aggregates the implicit feedback, they can only see the behaviour patterns, not the actual preferences. For example, if a user have bought an item it does not necessarily mean user liked the item. The might have bought the item out of necessity or on behalf of someone

else. Either way a user interacted with an item will not actually reflect the preference of a user. Finally, explicit feedback data indicates the users' preferences more clearly using different scales, whereas, in implicit feedback, the numerical value represents the confidence of interaction based on a user's actions. For example, a TPSP can be confident that a user who has bought an item five times might continue to buy it in the future. Thus, indicating the confidence levels of user preferences among items is an excellent approach to differentiate preference from consumption in implicit feedback.

To account for these various characteristics of implicit feedback, a confidence parameter c_{ij} is used to measure the preference of user i towards an item j (Hu et al., 2008). The confidence parameter is defined as:

$$c_{ij} = 1 + \alpha r_{ij}$$

If r_{ij} is a higher value, then the confidence parameter gives a stronger indication that the user i indeed prefers the item j. From experiments, it is found that higher accuracy is obtained when setting the value of $\alpha = 40$ (Hu et al., 2008).

The modified loss function for the MF algorithm based on implicit feedback is given as (Hu et al., 2008):

$$\min_{X,Y} \sum_{i,j} c_{ij}(p_{ij} - x_i^T y_j)^2 + \lambda(\sum_i ||x_i||^2 + \sum_j ||y_j||^2) \tag{5.2}$$

The partial derivatives of this loss function with respect to vector x_i are given as:

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= -2 \sum_j c_{ij}(p_{ij} - x_i^T y_j) y_j + 2\lambda x_i \\ &= -2 \sum_j c_{ij}(p_{ij} - y_j^T x_i) y_j + 2\lambda x_i \end{aligned} \tag{5.3}$$

The user latent factor is then updated as follows:

$$x_i^{t+1} = x_i^t - \gamma \frac{\partial L}{\partial x_i}, \tag{5.4}$$

where $\frac{\partial L}{\partial x_i}$ is computed using Eq. (5.3) and γ is the learning rate. Similarly partial derivatives of

this loss function with respect to vector y_j are given as:

$$\frac{\partial L}{\partial y_j} = -2\sum_i c_{ij}(p_{ij} - x_i^T y_j)x_i + 2\lambda y_j \tag{5.5}$$

The item latent factor is then updated as follows:

$$y_j^{t+1} = y_j^t - \gamma \frac{\partial L}{\partial y_j}. \tag{5.6}$$

5.2 Federated Matrix Factorization for Implicit Feedback

In the federated learning paradigm, users do not have to reveal their data to the untrustworthy service provider. Instead, they train the model locally and send the updated model parameters to the TPSP. The TPSP aggregates these model parameters from each user and updates the global model accordingly. This updated model is then returned to the user. Updating the local and global model and sharing model parameters is repeated until conditions for convergence are met. The MF-based recommendation system proposed in the previous section allows the TPSP to aggregate users' implicit feedback, which raises privacy concerns among users. To tackle this problem, a recently proposed federated collaborative filtering method (Ammad-Ud-Din et al., 2019) distributes the MF method introduced in the previous section so that the user's implicit feedback never leaves their own devices.

First, the TPSP initialises an item latent factor matrix Y, and each user initialises their user latent factors x_i locally. The users then download the item latent factor matrix from the server to compute their closed-form updated user latent factor x_i using Eq. (5.3) and Eq. (5.4). Formally y_j is updated on the TPSP-side using Eq. (5.5) and Eq. (5.6). However, Eq. (5.5) contains a component which sums the values aggregated from all the users. We, therefore, define this user-based stochastic gradient update component for the item latent factor matrix as:

$$\nabla y_{ij} = c_{ij}(p_{ij} - x_i^T y_j)x_i \tag{5.7}$$

All the users calculate ∇y_{ij} locally using their data and then report back ∇y_{ij} for all the items $\{0, 1, ...m\}$ to the TPSP. The TPSP then aggregates all these stochastic gradient components and

computes the stochastic gradient as follows:

$$\frac{\partial L}{\partial y_j} = -2 \sum_i \nabla y_{ij} + 2\lambda y_j$$

Finally, item specific stochastic gradient updates are then used to update the item latent factor at TPSP's side as follows.

$$y_j^{t+1} = y_j^t - \gamma \frac{\partial L}{\partial y_j}.$$

This proposed federated collaborative filtering method alternate between obtaining user latent factors x_i and then the item latent factor matrix Y. Algorithm 5 details this federated learning-based matrix factorization for implicit feedback.

Algorithm 5 Federated Matrix Factorization for Implicit Feedback

Input to the system : Individual user latent factor x_i^0 and Item latent factor Y^0
Output of the system: Converged Individual user latent factor x_i^* and Item latent factor Y^*
TPSP initializes item latent factor Y^0
User i initializes user latent factor x_i^0 and downloads item latent factor from TPSP Y^0
for iterations $t \in \{0, 1, 2....k\}$ do
  for $i \in \{0, 1, 2....n\}$ do.
    Download Y^t
    Update x_i^t
    for $j \in \{0, 1, 2....m\}$ do.
      Compute item latent factor stochastic gradient component ∇y_{ij}
    end.
  end
  TPSP aggregates the stochastic gradient components $\nabla y_{ij}^{\{0,1,2....m\}}$ from all users
  TPSP computes the master stochastic gradient update for the item latent factor $\frac{\partial L}{\partial y_j} = -2 \sum_i \nabla y_{ij} + 2\lambda y_j$
  TPSP updates $y_j^{t+1} = y_j^t - \gamma \frac{\partial L}{\partial y_j}$
end

5.3 LDP-based Federated Learning Matrix Factorization for Implicit Feedback

Federated matrix factorization allows users not to share their implicit feedback with the TPSP. However, each user sends the stochastic gradients of item latent factors to the TPSP. This practice raises privacy concerns as the TPSP can still learn about users' implicit feedback on items through

these stochastic gradients (Chai et al., 2020). Hence, we propose an LDP-based recommendation framework to tackle the issue of information leakage through gradients in federated matrix factorization for implicit feedback. Fig.19 illustrates our proposed recommendation framework.

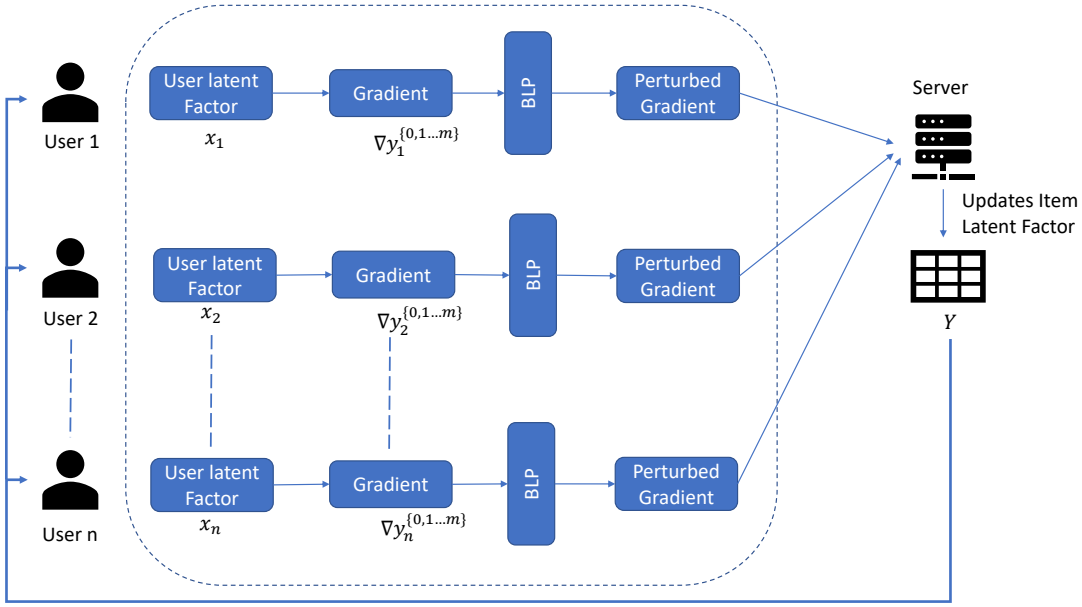


Figure 19: LDP-based Federated Matrix Factorization for Implicit Feedback

First, each user randomly initialises their user latent factor x_i^0 locally on their device, and they do not send this data to the TPSP. Then they download the global item latent factor Y^0 from the TPSP before commencing the model training. Both current user latent factor x_i^t and item latent factor Y^t are used to compute the updated user latent factor x_i^{t+1}. Then the user perturb the item latent factor stochastic gradient component ∇y_{ij} using the iterative input perturbation mechanism BLP. These perturbed stochastic gradient components are then aggregated by the TPSP and used to update the item latent factor matrix Y^{t+1}, which is then sent to the users for another round of training.

LDP Stochastic Gradient Perturbation

We use BLP as an iterative input perturbation mechanism in the federated matrix factorization to preserve the user's privacy. The BLP perturbation mechanism neglects any output values that do not fall within a given domain and samples noise until it produces an output value within the

predefined domain. We perturb ∇y_{ij} using BLP as shown below:

$$\nabla y'_{ij} = c_{ij}((p_{ij} + n_{ij}) - x_i^T y_j)x_i$$

where n_{ij} is the noise sample drawn from BLP and $\nabla y'_{ij}$ is the resulting perturbed stochastic gradient component. During each iteration, we use BLP to draw a noise sample and add that to the implicit feedback p_{ij}. Hence, at each iteration, the user produces a perturbed stochastic gradient. BLP mechanism calibrates the magnitude of noise added to original implicit feedback according to the sensitivity given by $\Delta f = 1$.

The definition of BLP is given by Definition 5. From this definition we define ΔC as:

$$\Delta C = \frac{C(l + \Delta f)}{C(l)}.$$

We proved that when the scale parameter $b \geq \frac{\Delta f}{\varepsilon - log\Delta C}$, it is sufficient to show that BLP mechanism W satisfies ε-local differential privacy (refer to Theorem 1 for proof). This proof demonstrates that BLP cannot satisfy ε-local differential privacy when inheriting the scale parameter from the Laplace mechanism. We now derive a sufficient condition for BLP to satisfy ε-local differential privacy when perturbing a stochastic gradient in a MF-based recommendation system.

Theorem 2. *The scale parameter $b \geq \frac{1}{\varepsilon}$ is sufficient to show that the Bounded Laplace mechanism W satisfies ε-local differential privacy when perturbing stochastic gradient in a MF-based recommendation system.*

Proof. Assume that e and e' are a pair of possible inputs to a Bounded Laplace mechanism and $e' = e + z$. Let $0 \leq z \leq \Delta f$. e^* represents a perturbed output produced by the BLP mechanism. Given the domain of the perturbed output is $[0, 1]$, we can note that,

$$Pr(W(e) \in [0, 1]) = \frac{1}{C(e)} Pr(M(e) \in [0, 1]),$$

where M represents the Laplace mechanism.

The sufficient condition under which W satisfies ε-local differential privacy is (refer to Theorem

1 for proof),

$$b \geq \frac{\Delta f}{\varepsilon - \log(\Delta C)}.$$

We use BLP as a stochastic gradient perturbation mechanism in our recommendation system. The stochastic gradient perturbation mechanism calibrates the magnitude of noise added to original ratings according to the sensitivity given by $\Delta f = 1$.

We define ΔC as:

$$\begin{aligned}\Delta C &= \frac{C(l + \Delta f)}{C(l)} \\ &= \frac{1 - \frac{1}{2}(e^{-\frac{\Delta f}{b}} + e^{-\frac{u - \Delta f - l}{b}})}{1 - \frac{1}{2}(1 + e^{-\frac{u-l}{b}})}.\end{aligned}$$

When $u = 1, l = 0$ and $\Delta f = 1$,

$$\Delta C = \frac{1 - \frac{1}{2}(1 + e^{-\frac{(1)}{b}})}{1 - \frac{1}{2}(1 + e^{-\frac{(1)}{b}})} = 1.$$

Thus $\log \Delta C = 0$.

Therefore we can conclude that a sufficient condition needed for the BLP mechanism to satisfy ε-local differential privacy in federated matrix factorization for implicit feedback recommendation system can be given by:

$$b \geq \frac{\Delta f}{\varepsilon},$$

or equivalently,

$$b \geq \frac{1}{\varepsilon}.$$

□

Additionally, the sequential composition property of differentially private mechanisms bounds the total privacy budget when reporting multiple perturbed outputs using the same input data (Dwork, 2008). For example, if the privacy budget used in each iteration of an algorithm is ε_k, then the privacy budget of the overall algorithm with k iterations is $\sum_k \varepsilon_k$. In federated learning the training phase for each user involves k number of iterations known in advance. If we set the privacy

budget for each iteration to ε, then the sequential composition property ensures that the privacy budget for the entire algorithm is $k\varepsilon$.

Algorithm 6 describes the LDP-based gradient perturbation mechanism.

Algorithm 6 LDP-based Stochastic Gradient Perturbation mechanism

Input to the Mechanism: Error value e_{ij}
Output of the Mechanism: Perturbed error value e'_{ij}
Compute $e_{ij} = p_{ij} - x_i^T y_j$
Add noise to original error value to obtain a perturbed error value: $e'_{ij} = e_{ij} + Lap(0, b)$
Set perturbed stochastic gradient component as $\triangledown y'_{ij} = c_{ij} e'_{ij} x_i$
If $(e'_{ij} \in (0, 1))$:
  Perturbed stochastic gradient component is set to $\triangledown y'_{ij}$
else
  repeat Step 4 until $(e'_{ij} \in (0, 1))$
Return Perturbed stochastic gradient to the TPSP

The output of Algorithm 6 is sent to the TPSP, who aggregates all the perturbed stochastic gradient components from each user to update the global item latent factor matrix.

Federated Learning Algorithm

The complete LDP-based federated matrix factorization algorithm is given in Algorithm 7.

Algorithm 7 LDP-based Federated Matrix Factorization for Implicit Feedback

Input to the system : Individual user latent factor x_i^0 **and Item latent factor** Y^0
Output of the system: Converged Individual user latent factor x_i^* **and Item latent factor** Y^*
TPSP initialises item latent factor Y^0
User i initialises user latent factor x_i^0 and downloads item latent factor from the TPSP Y^0
for iterations $t \in \{0, 1, 2....k\}$ do
  for $i \in \{0, 1, 2....n\}$ do.
    Download Y^t
    Update x_i^t
    for $j \in \{0, 1, 2....m\}$ do.
      Compute perturbed item latent factor gradient $\triangledown y'_{ij}$ using Algorithm 6
    end.
  end
  TPSP aggregates the gradient components $\triangledown y_{ij}^{\{0,1,2....m\}}$ from all users
  TPSP computes the master gradient update for the item latent factor $\frac{\partial L}{\partial y_j} = -2 \sum_i \triangledown y_{ij} + 2\lambda y_j$
  TPSP updates $y_j^{t+1} = y_j^t - \gamma \frac{\partial L}{\partial y_j}$
end

5.4 Evaluation

5.4.1 Datasets

We use two datasets:Movielens (25 million ratings) (Harper and Konstan, 2015) and Jester (Goldberg et al., 2001) in the evaluation. Table 8 provides a detailed view of the datasets.

Table 8: Datasets for Federated Learning Evaluation

Dataset	Total Ratings	No of Items	No of Users	Rating Scale
Movielens	25 Million	62,000	162,000	0.5 to 5
Jester	2 Million	100	73,421	-10 to 10

We converted each rating in these datasets to either 1 and 0 value to create an implicit feedback-based dataset. For both datasets the implicit feedback p_{ij} is generally derived from the rating r_{ij} given by the user i on item j as:

$$p_{ij} = \begin{cases} 1, & \text{if } r_{ij} > 0, \\ 0, & \text{if otherwise.} \end{cases}$$

To evaluate the impact of user/item set size on the performance of the system, we set the size of the training group as indicated in Table 9.

Table 9: User/Item Set Size for Movielens and Jester Datasets

Population Size	Item size (Movielens)	Item size (Jester)	User size
Small	1000	10	1000
Medium	5000	50	10000
large	10000	100	50000

5.4.2 Evaluation Metric

For privacy budget ε, we consider the value range from 0.1 to 3; lower values of privacy budget ε guarantee stronger privacy protection for users. When it comes to implicit feedback-based recommendation systems, predicting which item a user will interact with is essential rather than a rating. Therefore, ranking-based performance evaluation metrics such as Hit Ratio (HR) (He et al., 2016) is more suitable for such systems. $HR@k$ measures whether the test item is present on the top-k

recommended items. $HR@k$ is computed as follows:

$$HR@k = \sum_{i \in z_{test}} \frac{I(rank(i,j) \leq k)}{|z_{test}|},$$

where I is the indicator function which returns 1 if the test item is found in user i's recommended items and 0 otherwise, z_{test} is set of test items and $rank(i,j)$ is the user i's recommended list. Intuitively the HR value is calculated as follows:

- Generated top-k recommended items to all the users in the test dataset z_{test}
- If an item which the user has already interacted with appears in the top-k of the recommended items, it is a hit.

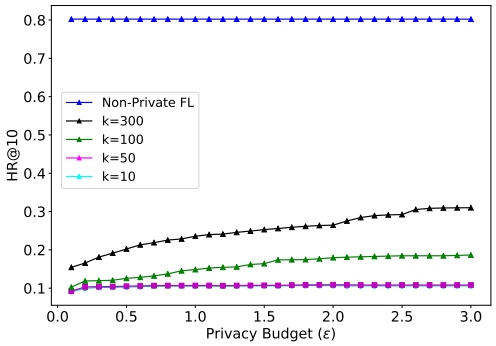
We evaluate the performance of our proposed recommendation system using the hit ratio metric (Hit@10).

5.4.3 Results

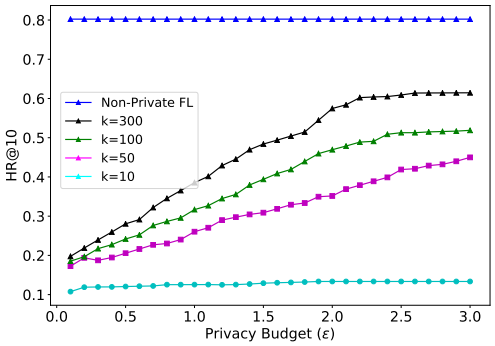
We compare our model with a baseline non-private implicit feedback-based recommendation system. The non-private recommendation model does not use any gradient perturbation mechanism and uses MF as the recommendation model. We use the whole dataset to compute the hit ratio for this model. The baseline method provides an upper bound for the hit ratio value. This section discusses the results of the empirical evaluations we carried out using Movielens and Jester datasets. We first evaluate the relationship between utility and privacy budget ε. Then we assess the relationship between the utility and the number of iterations k used to learn the global and local model parameters. We also look at the impact of user/item set size on utility.

Privacy-Utility Trade-off

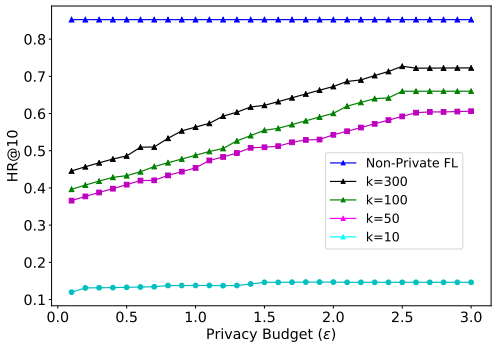
Figure.20 and Fig.21 illustrate the utility of our proposed recommendation systems measured by the metric hit ratio (HR@10) for Movielens and Jester datasets. We vary the privacy budget ε from 0.1 to 3. As expected, the hit ratio score increases when the privacy budget increases. Because when privacy budget ε increases, the magnitude of privacy loss the BLP mechanism permits increases, which causes a rise in the hit ratio metric. This trend is similar across all the user/item set sizes for both datasets and a different number of iterations.



(a) Movielens Small



(b) Movielens Medium

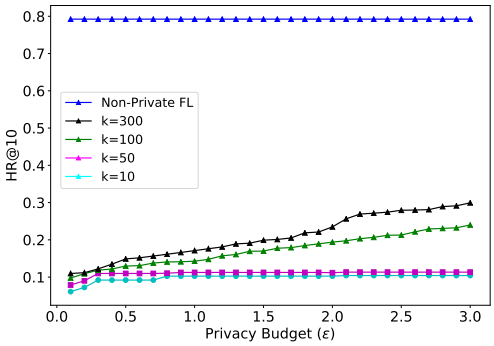


(c) Movielens Large

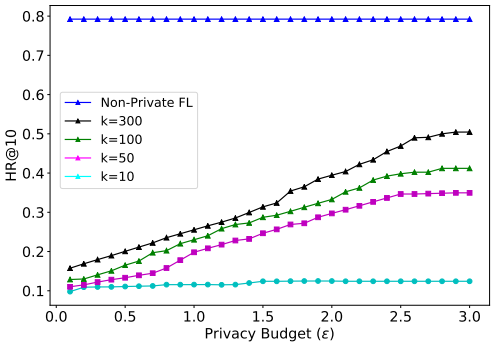
Figure 20: HR@10 When Varying Privacy Budget for Small, Medium and Large Movielens Dataset

Utility and Number of Iterations

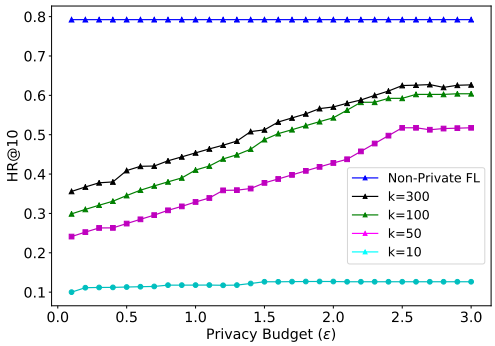
Fig.22 demonstrates the variation in utility of our proposed recommendation system when the number of iterations k are set to $10, 50, 100$ and 300. We use a medium user/item set while keeping the privacy budget at $\varepsilon = 3$ to observe the variation. As expected, utility is low when the number of iterations is low because fewer iterations are insufficient for the recommendation systems to learn converged item and user latent factors. However, we observe a significant increase in hit ratio values for both Movielens and Jester datasets with an increasing number of iterations. This increase demonstrates that when the number of iterations is high, the system can effectively learn converged user and item latent factors. However, increasing iterations impact the communication cost during the training period as the global and local parameters are continuously shared between the TPSP and the selected group of training users. Hence, the number of iterations has to be justified in terms of utility gain.



(a) Jester Small



(b) Jester Medium

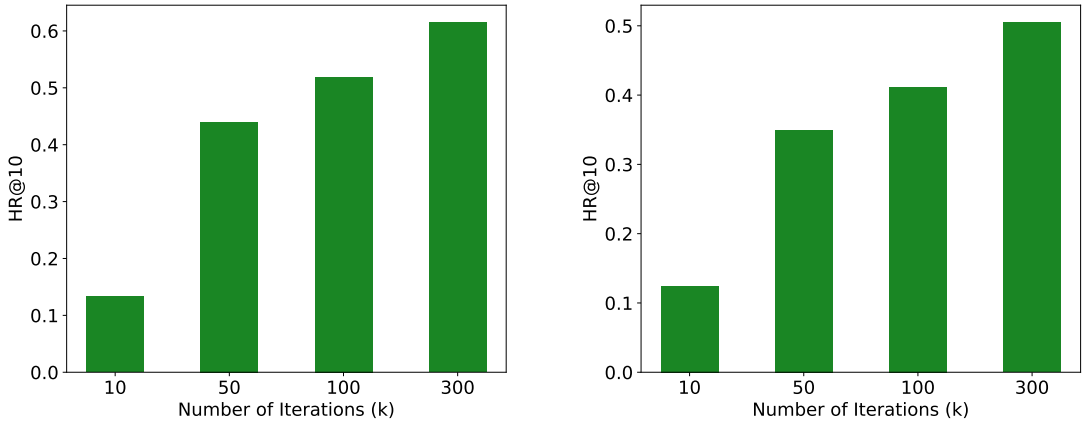


(c) Jester Large

Figure 21: HR@10 When Varying Privacy Budget for Small, Medium and Large Jester Dataset

Utility and Size of the Training Dataset

We use three different data groups to demonstrate the effect of user-item set size on utility metrics in Fig.23. We observe approximately a 50% increase in the utility when the size of the Movielens dataset increases from small to medium and only a 10% increase in the utility when the size of the dataset increase from medium to large. Similarly, we observe 50% and a 20% increase in the utility with similar settings for the Jester dataset. These increases are observed when the privacy budget is set to $\varepsilon = 3$, and the total number of iterations is set to $k = 300$. From this trend, we can observe that increasing the number of users from medium to large does not necessarily yield higher accuracy. Because when the number of items increases, the number of perturbed stochastic gradients reported by each user increases. Consequently, more perturbed stochastic gradients are sent to the TPSP, which negatively impacts the overall performance of the recommendation system.



(a) Movielens

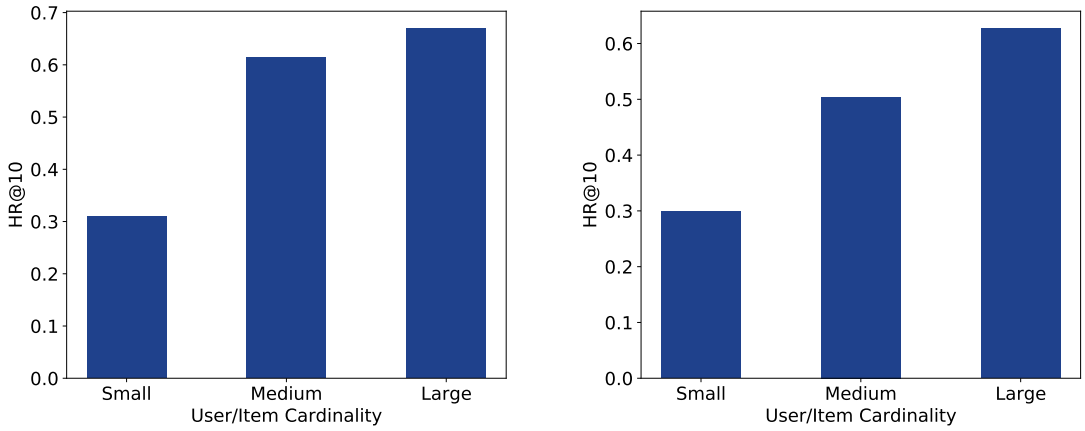
(b) Jester

Figure 22: HR@10 When Varying Number of Iterations for Movielens and Jester Dataset. Privacy Budget $\varepsilon = 3$ and User/Item Set Size is Medium.

5.5 Discussion

This chapter presents an LDP-based federated matrix factorization for implicit feedback where we perturb stochastic gradients using BLP as an iterative input perturbation mechanism. Through empirical evaluations, we observe that our proposed system's utility depends on privacy budget ε, the number of iterations k and the size of the user/item set. We note that perturbing stochastic gradient using the BLP mechanism does not yield significant utility for small user/item set size and lower iterations. Even though increasing the size of the user/item set and the number of iterations causes an increase in utility, it requires the users to communicate more perturbed stochastic gradients to the TPSP, which is undesirable due to privacy and communication limitations. Additionally, such an approach requires the users to be available online until the training phase is completed, which is impractical in real-world scenarios. Hence, investigating the suitability of the federated reconstruction paradigm would be an interesting avenue. We can also motivate our solution further via model-agnostic meta-learning, allowing users to learn user/item latent factor matrices more efficiently than regular federated learning models.

We also observe that compared to explicit feedback, implicit feedback is easier to aggregate. However, negative feedback representation is a problem that needs to be tackled to represent a user's preferences accurately using implicit feedback. Our method uses the popular solution of modelling all the missing explicit feedback as negative feedback (Hu et al., 2008). Yet, this approach impairs



(a) Movielens

(b) Jester

Figure 23: HR@10 Performance for Movielens and Jester Datasets of Varying User/Item Set Size. Privacy Budget $\varepsilon = 3$ and Number of Iterations $k = 300$.

the utility by considering the missing data as observed data. Hence, we would like to investigate new methods such as incorporating item popularity with confidence parameter (c_{ij}) to represent negative implicit feedback. Additionally, we used MF as the approach to learning user/item latent matrices from the implicit feedback. Despite MF being an effective method for explicit feedback, it can be improved to support implicit feedback. The inner product used to predict a user's missing ratings may not be sufficient to capture the complex structure of a user's implicit feedback. Hence we would like to explore the use of neural collaborative filtering to tackle our problem and enhance the utility further.

We consider our work the first step toward privacy-preserving federated MF for e-commerce systems. We aim to explore multiple applications of this model in the future it can be used on social networking sites to predict users' preferences too. Aggregating implicit feedback on social networking sites will yield a more accurate representation of a user's implicit feedback than interpreting it using explicit feedback. Hence, we also would like to evaluate the performance of our model using implicit feedback datasets directly obtained from these social networking sites.

5.6 Conclusion

To the best of our knowledge, we present the first approach to introduce a privacy-protecting federated learning-based recommendation system for implicit feedback. We have empirically eval-

uated the privacy-utility trade-off in our system using three different sizes of user/item sets for Movielens and Jester Datasets. We proved that the utility of our recommendation system relies on user/item set size, the number of iterations used for training and the privacy budget ε. Even though the recommendation system's utility can be further improved, we have shown that our proposed recommendation system is suitable for providing strong privacy protection to users.

Chapter 6

Untraceable Payment System

Third-party service provider (TPSP) owned mobile payment systems allow customers to pay for their transactions in a fraction of a second. However, they do not protect a customer's privacy in the face of a TPSP. Customers cannot make transactions anonymously since the TPSP collects all the transaction-related information. Hence, it is vital to ensure that a customer's transactions are untraceable on the face of a TPSP. This chapter proposes a TPSP-based untraceable mobile payment system to address the shortcomings identified in other works. The TPSP is considered untrustworthy as it aggregates all the transaction details and infers sensitive information about the users. However, the role of the TPSP is still essential in the proposed system to ensure that malicious adversaries cannot undermine any legal controls or carry out illegal transactions.

We also introduce a double-spending detection mechanism to reveal the identity of the dishonest customers to the TPSP for further actions in our system. Additionally, the system ensures that the merchants cannot act dishonestly during the execution of the protocol, and the corresponding transaction fails as a result if they try to perform malicious activities such as double-spending. The proposed untraceable payment system satisfies security and privacy requirements such as untraceability, double-spending detection, exculpability, confidentiality, authenticity and unforgeability. We formally analyse our untraceable payment system using the automated verification tool Proverif. We verify the following characteristics of the payment system using Proverif:

- Secrecy - A malicious adversary cannot obtain or eavesdrop on the messages transmitted between the main actors in the protocol. Secrecy ensures the confidentiality of the commu-

nication among participants of the protocol.

- Message Authentication - This property guarantees the authenticity of the participants in the protocol.
- Untraceability - This property enables customers to use the payment system without disclosing any transaction details to the TPSP.

Through theoretical analysis, we also show that the following security properties are satisfied in our system:

- Exculpability - We consider a dishonest customer or merchant in our analysis. This property guarantees that neither customer nor the merchant can double-spend the payment token during the execution of the protocol. We also show how our protocol reveals the identity of the dishonest customer if they try to double-spend the token.
- Unforgeability - During the execution of the protocol, we ensure that only the TPSP can issue a valid payment token. No one else can issue a valid payment token or forge the signature of the TPSP on the payment token.

Apart from these security and privacy requirements, another main requirement for untraceable payment systems is to incur as little computational cost as possible on the customer side. The cryptographic primitives used in our proposed system are lightweight and more suitable for mobile devices. We show that the computational cost incurred in our system is low compared to the payment systems proposed by (Baseri et al., 2013) and (Deya et al., 2012).

6.1 Third-Party based Payment Systems

TPSP-based payment systems have become immensely popular among users. Users can make in-store and online payments conveniently using their smartphones, and these systems are an integral part of an e-commerce platform. Generally, a TPSP who owns these payment systems acts as an intermediary between customers and merchants. If there is a dispute between a customer and a merchant, TPSP solves the dispute to ensure a fair exchange of goods or services. The most popular TPSP owned payment systems are Google Pay, Apple Pay, Ali Pay, PayPal and Venmo.

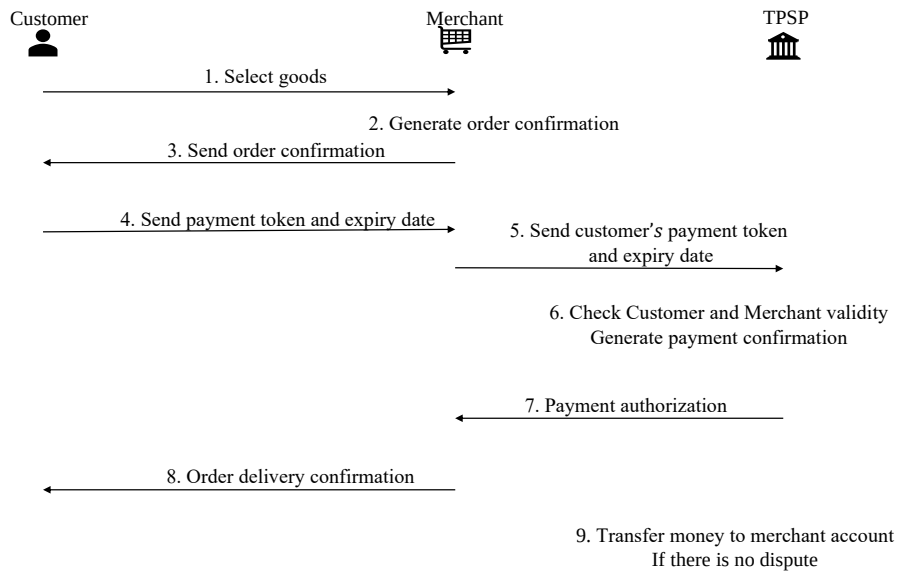


Figure 24: Workflow of the Generic TPSP-based Mobile Payment System

Fig.24 illustrates the workflow of a TPSP owned payment system. Both customers and merchants should register with the TPSP before using the services offered by these systems. A transaction between a customer and a merchant consists of the following stages:

- Tokenizing Card: After registering with the TPSP, customers add their card details to the payment system. These card details are then converted into a payment token and stored in the customer's mobile device for Near Field Communication (NFC) based payments. The token information is also shared with the TPSP and stored as part of customer details. Whenever a customer purchase products or services of interest, it sends the order details to the merchant. After receiving the order details from the customer, the merchant generates an order confirmation which indicates the total cost for the order.
- Paying the Merchant: Once the customer receives the order confirmation from the merchant, they tap the mobile device on the merchant's point of sale terminal. The customer's mobile device sends the payment token and expiry date to the merchant through the NFC protocol.
- Merchant Payment Processing: The merchant then uses the token details to process the payment through the TPSP.
- TPSP translating the payment token: Initially, the TPSP validates the token details received

from the merchant. If the validation is successful, it translates the token information to an actual card number. TPSP uses this card number to identify the customer's account. TPSP then subtracts the order amount from the customer's account and transfers the same amount to a temporary holding account. Then it sends an authorisation response to the point of sale terminal to indicate whether the transaction is successful or not. The money is not deposited to the merchant's account immediately until the customer receives the product and there are no disputes detected between the customer and the merchant.

- Terminal Notification: The point of sale terminal notifies both the merchant and the customer whether the payment attempt is successful.

6.2 An Untraceable Third-party based Payment System

6.2.1 Overview of the System

In this section, we present our untraceable TPSP-based payment system. We have three actors in our system: a customer (C), a merchant (M) and a third-party service provider ($TPSP$). The payment system consists of six stages: user registration, purchase request, token withdrawal, payment by token, token deposit and refund. In the user registration stage, users (both customer and merchant) register with a TPSP. During the purchase request stage, a mutual zero-knowledge authentication protocol is used between the customer and merchant to verify whether they are valid registered users of a TPSP without requiring any assistance from the TPSP.

After successful mutual authentication, the system establishes a secure session between the customer and the merchant for further communication. Whenever a customer needs to make a payment, it withdraws a valid payment token from the TPSP in the token withdrawal stage. The customer uses this token to pay for the product or services they purchased from the merchant during the payment by token stage. In the token deposit stage, the merchant sends the payment token received from the customer to the TPSP, who verifies the validity of the payment token. TPSP sends a payment authorisation message to the merchant if the verification is successful. The refund stage is optional and executed only if the customer is not happy with the received product or service. Fig. 25 illustrates the main stages involved in our untraceable payment systems.

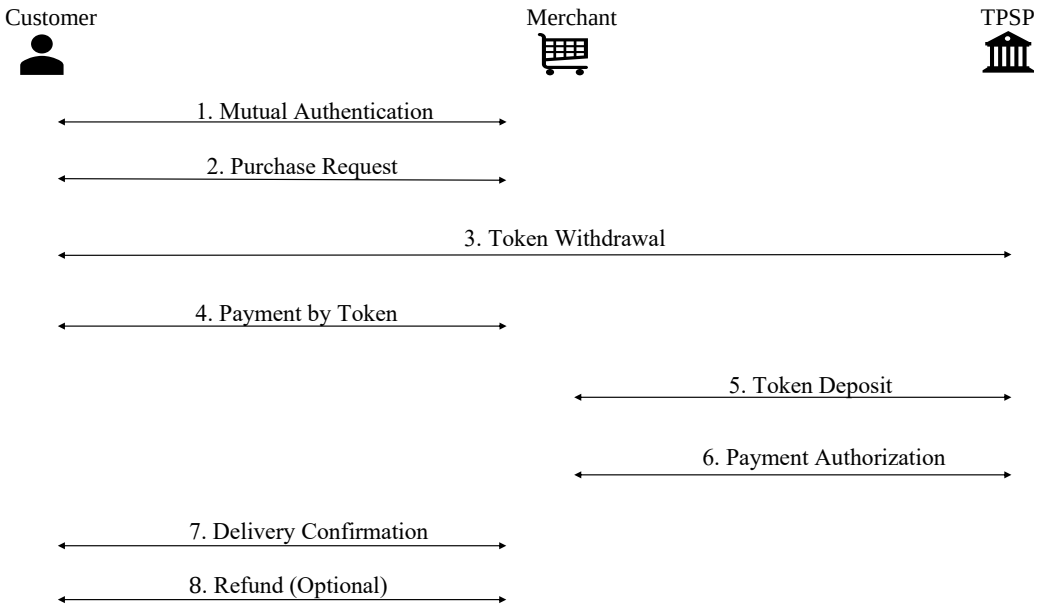


Figure 25: Workflow of the Proposed Untraceable Payment System

6.2.2 Registration

The customer and the merchant must first register with the TPSP to use their services. The TPSP generates an RSA key pair (public key P_k and private key S_k) for each registered user. Additionally, long term identity key-pairs (public identity keys $V_{m,j}$ and private identity keys $S_{m,j}$) are generated at the user-side and stored locally. The private identity keys $S_{m,j}$ are kept secret, and the user shares the public identity keys $V_{m,j}$ with other registered users during the mutual authentication phase. The information about these identity keys is not revealed to the TPSP at any stage. The zero-knowledge proof-based mutual authentication mechanism uses this identity key pair. The key generation process is as given below:

- TPSP chooses two large prime numbers p_z and q_z and computes $N = p_z \times q_z$. The prime numbers p_z and q_z are kept private, and N is shared publicly.
- Each registered user generates k number of public and private identity key pairs locally on their device where $k = 1, 2, ..., 20$. We use $V_{m,j}$ and $S_{m,j}$ to represent the group of public identity keys and the private identity keys generated by the user m respectively. The public and private key generation are as follows:
  - Generation of private keys: Randomly generate an integer $S_{m,1}$ such that $1 \leq S_{m,1} \leq$

$N-1$

$$S_{m,j} = S_{m,1} - j + 1 \ (2 \leq j \leq k).$$

– Generation of public keys:

$$V_{m,j} = \frac{1}{S_{m,j}^2} mod\ N \ (1 \leq j \leq k).$$

Since users generate their identity key-pairs at their end, they need to send the public identity keys $(V_{m,j})$ to the TPSP for validation. To avoid revealing any information about the public identity keys to the TPSP, users send only the blinded public identity keys. We use the blind signature scheme proposed by Zhang and Kim (Zhang and Kim, 2003) to blind the public identity keys.

The TPSP has to set certain system parameters for the blind signature scheme. Let P be the generator of an additive cyclic group $G1$ whose order is prime q. G_2 is a multiplicative cyclic group of the same order q. The bilinear pairing e is given by $e : G1 \times G1 \rightarrow G2$. The TPSP chooses a random number s as the master secret key that is known only to the TPSP and sets a system parameter $P_{pub} = sP$. Then the TPSP chooses two hash functions, $H1$ and $H2$.

Fig. 26 illustrates the public identity key validation process that takes place between a registered user and the TPSP. The detailed description is listed below:

- TPSP shares the system parameters $\{G1, G2, e, q, P, P_{pub}, H1, H2\}$ with each registered user and keeps master secret key s private.
- The TPSP then generates a public key $Q_{ID} = H2(ID)$ and private key $S_{ID} = sQ_{ID}$.
- TPSP chooses a random number $r \in \mathbb{Z}_{\shortparallel}$ and computes a commitment value $U = rQ_{ID}$. The TPSP then sends U to the user.
- The user then chooses blinding factors $\alpha, \beta \in \mathbb{Z}_{\shortparallel}$ and computes $U' = \alpha U + \alpha\beta Q_{ID}$. Then the user blinds the public identity keys $V_{m,j}$ by computing a blind message $h = \alpha^{-1} H1(V_{m,j}, U') + \beta$. The blind message h is then sent to the TPSP for validation. TPSP cannot unblind this message on their own. Hence they cannot infer anything about a user's public identity keys.

- TPSP computes $V = (r + h)S_{ID}$ and sends V back to the user.
- User computes $V' = \alpha V$.
- The TPSP's signature on the public identity keys is $(U', V', V_{m,j})$ which can be used by the user to prove that they have a validated public identity keys.

Users who need to verify the validity of the public identity keys $V_{m,j}$ of another user can do it using the public system parameters shared by the TPSP and public key Q_{ID}. The TPSP's signature is valid if and only if:

$$e(V', P) = e(U' + H1(V_{m,j}, U')Q_{ID}, P_{pub}). \tag{6.1}$$

If a user fails to validate their public identity keys, they are not be able to participate in any successful transactions in our payment system.

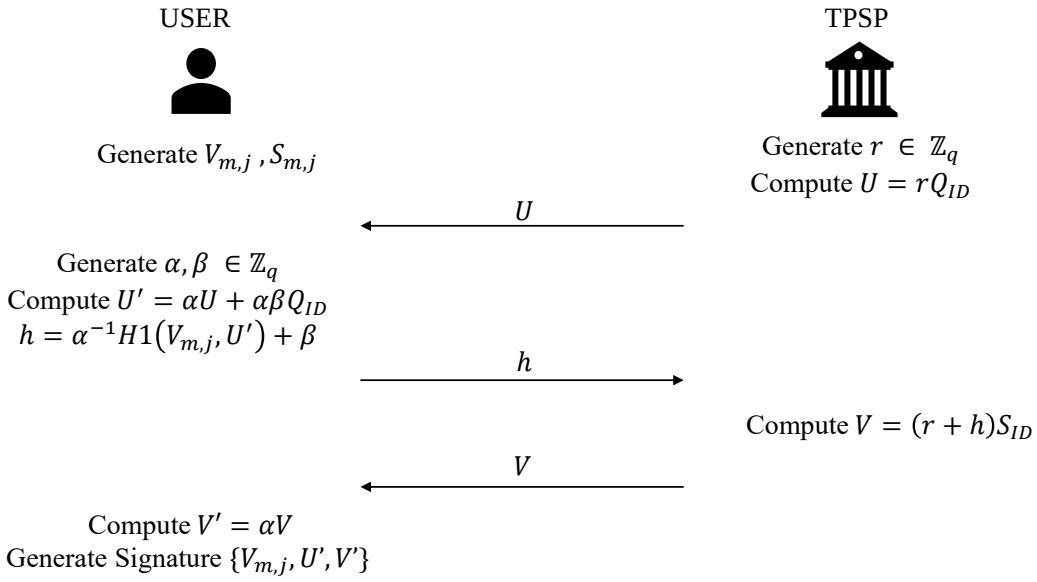


Figure 26: Public Identity Keys Validation using Blind Signature

6.2.3 ZKP Mutual Authentication

The registered users use a mutual authentication mechanism based on zero-knowledge proof among themselves to authenticate each other without any assistance from the TPSP. This mutual authentication process ensures that the TPSP cannot link a customer and merchant through the

authentication process. In our protocol, we modify the TinyZKP (Ma et al., 2014) authentication scheme and use it as a lightweight mutual authentication scheme. We modify the authentication scheme so that it does not require any assistance from the TPSP. We use the identity key pairs created during the registration stage as the public and private keys in the mutual authentication scheme.

In our scheme, both the customer and the merchant alternatively take the role of prover and verifier. When the customer or merchant wants to prove their identity to the other party, they act as the prover. Similarly, when they need to verify the identity of another customer or merchant, they act as the verifier. Our modified mutual authentication protocol ensures that the verifier cannot learn anything about the prover's respective $S_{m,j}$ (private identity key). Hence, the verifier cannot maliciously act as the prover to another verifier as they hold no knowledge about the prover's private identity keys. This method helps prevent the merchant from double-spending the payment token with another merchant. Because without a successful execution of the mutual authentication scheme, the merchant cannot carry out the payment by token stage. Fig. 27 illustrates our modified mutual authentication scheme. The detailed description is listed below:

- TPSP uses the parameters generated at the registration stage p_z, q_z and N in the mutual authentication scheme.
- The customer initially sends an authentication request message to the merchant requesting an authentication challenge.
- The merchant then generates a challenge $M_{chall} : (e_1, e_2, ...e_k)$ where $e_k = 0 \; or \; 1$ and $k = 1, 2, ..., 20$. The merchant then sends the challenge (M_{chall}) and a timestamp T_1 to the customer.
- After receiving the challenge, the customer selects a random number a such that $1 \leq a \leq N - 1$ and computes X_m and Y_m as follows:

$$X_m = a^2 \; mod \; N,$$

$$Y_m = a \prod_{j=1}^{k} S_{m,j}^{e_j} \; mod \; N \; (1 \leq j \leq k),$$

where $S_{m,j}$ are the private identity keys of user m.

- The hashed value $H(X_m)$ is computed and the customer sends $H(X_m)$, Y_m, public identity keys $V_{m,j}$, Q_{ID} along with the TPSP's signature (U', V') and a timestamp T_2 to the merchant.

- The merchant initially verifies whether $\Delta t > T$ where $\Delta t = T_2 - T_1$ and T is the threshold response time set by the merchant to indicate before when the authentication reply has to be sent by the customer. If this verification fails, the merchant rejects the authentication reply from the customer.

- If the verification is successful, then the merchant verifies whether the signature (U', V') is valid by performing a verification check using the system parameters issued by the TPSP using Eq. (6.1). If the verification fails, the merchant rejects the authentication reply from the customer.

- If the signature verification is successful, then the merchant computes X'_m using the customer's identity public key as follows:

$$X'_m = Y_m^2 \prod_{j=1}^{k} V_{m,j}^{e_j} \; mod \; N \; (1 \leq j \leq k).$$

  If $H(X'_m)$ is equivalent to $H(X_m)$ then the customer is authenticated successfully by the merchant.

After the merchant verifies the customer's identity, the mutual authentication process repeats to verify the merchant's identity by the customer. This time, the merchant acts as the prover, and the customer acts as the verifier. The payment protocol can progress to the next stage of initiating a purchase request if and only if the mutual authentication process is successful.

6.2.4 Purchase Request

After successful mutual authentication, the customer selects the product or service they intend to buy and sends a purchase request to the merchant. The detailed description of the purchase request stage is listed as below:

- First C sends a purchase request containing the list of product IDs $\{Item_1, ...Item_n\}$ to M.

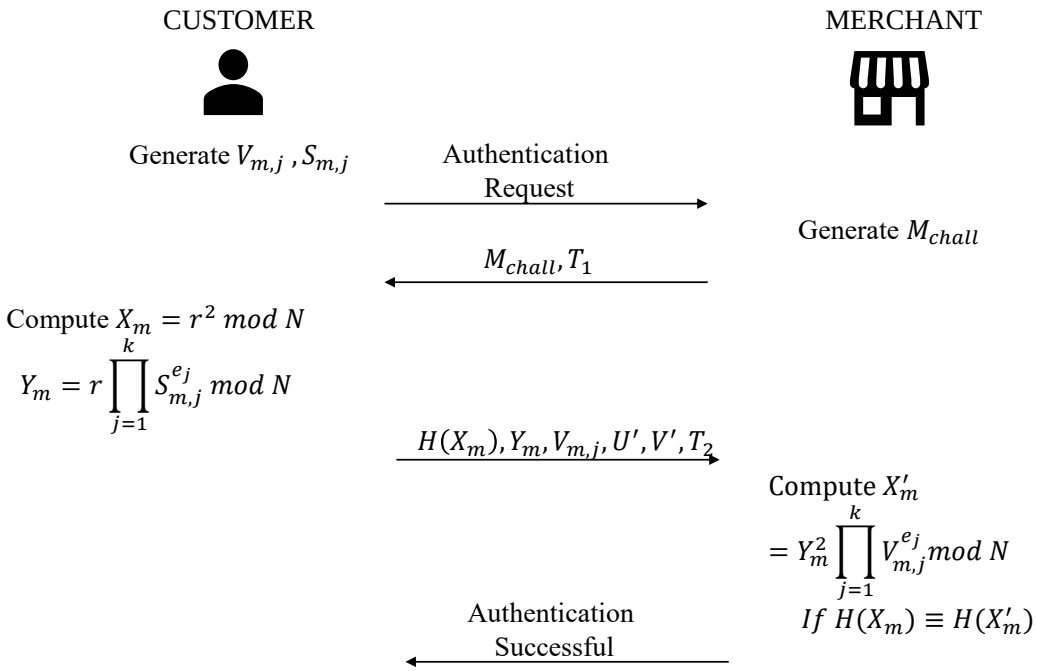


Figure 27: Mutual Authentication Scheme

- M generates a random number $R \in \mathbb{Z}_n$, accumulates the prices for the list of products as X and generates a transaction ID P_{ID}.
- M sends a purchase reply tuple $\{R, P_{ID}, X\}$ to C.

We use the random number R to generate a security tag for double-spending detection. The detailed description is given in section 6.2.7.

6.2.5 Token Withdrawal

We use a RSA-based partially blind signature scheme proposed (Chien et al., 2001) in our protocol as it incurs low computation cost at the user-side compared to other partially blind signature schemes. We use the product price X as the common information that can be revealed to the TPSP. Before token withdrawal, the TPSP chooses two large prime numbers p_t and q_t and computes the modulus $n_t = p_t.q_t$. Additionally the TPSP computes a totient function $\phi(n_t) = (p_t - 1)(q_t - 1)$ and private exponent d_t in such way $e_t.d_t = 1 \ mod \ \phi(n_t)$. TPSP publishes (e_t, n_t) and keeps the values of (d_t, p_t, q_t) private. TPSP also publishes a hashing algorithm $H3$. The detailed description of the token withdrawal stage is listed as below:

- C generates a random Token ID T_{ID} and randomly chooses two numbers r_t and u_t where $r_t, u_t \in \mathbf{Z_{n_t}}$.
- C blinds the Token ID T_{ID} as follows:

$$Y = r_t^{e_t} H3(T_{ID})(u_t^2 + 1) \; mod \; n_t,$$

where Y is the resulting blind message.

- C then sends (X, Y) to the TPSP.
- TPSP randomly chooses a positive integer a_t where $a_t < n_t$ and sends it to C.
- After receiving a_t, C randomly generate another random number r'_t and computes $b_t = r_t.r'_t$.
- C then computes $\beta = b_t^{e_t}(u_t - a_t) \; mod \; n_t$ and sends β to the TPSP.

6.2.6 Token Issuance

The token issuance stage is where the TPSP issues a valid signature on the payment token without knowing what is the actual value of the Token ID T_{ID}. The detailed process is as given below:

- After receiving β from C, TPSP computes $\beta^{-1} \; mod \; n_t$ and computes t as follows:

$$t = H3(X)^{d_t}(Y(a_t^2 + 1)\beta^{-2})^{2d_t} \; mod \; n_t.$$

- TPSP then sends (β^{-1}, t) to C.
- Upon receiving (β^{-1}, t), C acquires the signature of the TPSP on Token ID T_{ID} by computing:

$$c = (u_t a_t + 1).\beta^{-1}.b_t^{e_t} \; mod \; n_t,$$

and

$$s = t.r_t^2.r_t'^4 \; mod \; n_t. \tag{6.2}$$

C uses (X, T_{ID}, c, s) as the TPSP's signature on Token ID T_{ID}.

6.2.7 Payment By Token

C submits the validated token to M as the chosen product or service payment. C also produces a security tag as part of the double-spending detection process. We use this security tag to reveal the dishonest customer's identity if they try to double-spend the token. The pair of public (P_k) and private (S_k) keys generated by the TPSP for C during the registration stage is used to compute the security tag. The detailed process of the payment by token phase is as given below:

- C generates a security tag T_c using the random number R sent by M during the purchase request stage and the public key P_k. Generation of T_c is as follows:

$$T_c = P_k^{(1+R)}.$$

- C then sends a payment request tuple $\{T_{ID}, X, c, s, T_c\}$ to M.

6.2.8 Token Deposit

After receiving the payment request tuple from C, M verifies the signature on the Token ID and then forwards the payment tuple to TPSP during the token deposit stage. The token deposit stage follows the process as given below:

- M verifies whether the signature on the payment token is valid by computing:

$$s^{e_t} = H3(X)H3(T_{ID})^2(c^2+1)^2 \; mod \; n_t$$

  If the verification fails, then M sends **Payment Failed** message to C.

- If the verification is successful, M forwards the payment request tuple $\{T_{ID}, X, c, s, T_c\}$ to the TPSP and at the same time stores T_c, R, T_{ID} locally.

- TPSP verifies whether the signature on the payment token is valid. If the verification fails, then the TPSP sends **Token Deposit Failed** message to M.

- If the verification is successful, then the TPSP confirm whether the Token ID (T_{ID}) already exists in either the deposit or refund token table. If it exists, then the TPSP sends **Token Deposit Failed** message to M and commences the double-spending detection process.

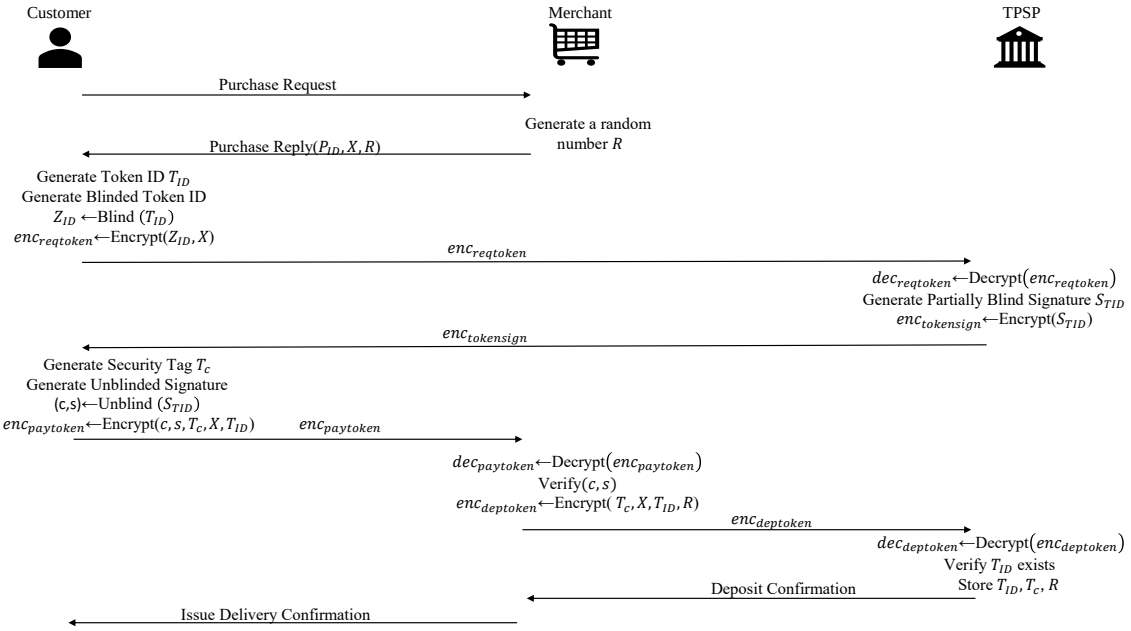


Figure 28: Overview of the Proposed Untraceable Payment System

- If the verification is successful, the TPSP sends **Token Deposit Successful** message to M.

If the token deposit is successful, the TPSP deposits the same denomination amount of money in a temporary holding account and stores the Token ID in the token deposit table. After receiving the deposit confirmation message from the TPSP, M sends a product delivery confirmation to the customer that marks the end of the transaction between C and M. TPSP transfers the money to the merchant's account after a specific period if there are no disputes between the customer and the merchant. Fig.28 illustrates purchase request, token withdrawal, token issuance, payment by token and token deposit stages.

6.2.9 Refunding

Our protocol introduces a refund mechanism that does not require any assistance from the TPSP. If the customer is not satisfied with a product or service, they can request a refund from the merchant. Instead of the customer, the merchant executes the token withdrawal and token issuance phases with the TPSP during the refund stage. They obtain a valid anonymous token with the appropriate denomination amount that can be used in the payment by token stage to refund the customer. The customer then executes the token deposit stage by forwarding the payment token tuple to the TPSP. After a customer successfully deposits the refund token, the TPSP includes the corresponding

refund Token ID in the refund token table.

6.3 Formal Proof for Security and Privacy Analysis

We verify the following properties of our untraceable payment protocol through theoretical analysis:

- Untraceability - A payment token is untraceable if no one can trace the link between a token ID and the customer.
- Unforgeability - A payment token is unforgeable if no one except the authorised TPSP can issue a valid payment token.
- Exculpability - This property ensures that the merchant cannot double-spend the token, and the identity of the customer who tries to double spend the token can be revealed.

6.3.1 Untraceability

We ensure that for a given valid unblinded signature (X, c, s) on the token ID T_{ID}, no one can link the signature to a customer's identity. Hence our proposed protocol does not reveal the link between a customer and a payment token on the face of a TPSP or merchant. This is the untraceability property we preserve in our protocol. This section shows that our untraceable payment protocol maintains this property. Assume TPSP stores a record of all the parameters they compute and receive from each customer during the token withdrawal and issuance phases. To ensure that the untraceability property is preserved during the protocol, we have to prove that Eq. (6.2) always holds valid for all the signature records stored by the TPSP.

Assume a valid signature tuple $\{X, T_{ID}, c, s\}$ of a customer and the corresponding signature record stored by the TPSP is represented by tuple $\{X, Y, a_t, \beta, t\}$. Let's assume that the TPSP is able to derive b_t, r_t and u_t in such way to satisfy the following equations:

$$Y = r_t^{e_t} H3(T_{ID})(u_t^2 + 1) \; mod \; n_t, \tag{6.3}$$

$$\beta = b_t^{e_t}(u_t - a_t) \; mod \; n_t \tag{6.4}$$

$$c = (u_t a_t + 1)(u_t - a_t)^{-1} \; mod \; n_t \tag{6.5}$$

Hence from Eq. (6.5) we can derive,

$$u_t = (ca_t + 1)(c - a_t)^{-1} \; mod \; n_t \tag{6.6}$$

If we substitute Eq. (6.6) into Eq. (6.3), then we can have,

$$Y = r_t^{e_t} H3(T_{ID})(((ca_t + 1)(c - a_t)^{-1})^2 + 1) \; mod \; n_t.$$

Then r_t can be derived as,

$$r_t = Y^{d_t} H3(T_{ID})^{-d_t} (u_t^2 + 1)^{-d_t} \; mod \; n_t \tag{6.7}$$

If we substitute Eq. (6.6) into Eq. (6.4), then we can have,

$$\beta = b_t^{e_t}((ca_t + 1)(c - a_t)^{-1} - a_t) \; mod \; n_t$$

Then b_t can be derived as,

$$b_t = \beta_t^d (u_t - a_t)^{-d_t} \; mod \; n_t \tag{6.8}$$

If we substitute the values of b_t and r_t in Eq. (6.2), it still holds. However, the TPSP cannot link the records of a customer to the unblinded signature because Eq. (6.2) holds for the records of all

the customers. We can show this by computing Eq. (6.2) by using b_t and r_t as follows:

$$
\begin{aligned}
s &= t.r_t^2.r_t'^4 \; mod \; n_t \\
&= H3(X)^{d_t}(Y(a_t^2+1)\beta^{-2})^{2d_t} r_t^{-2} b_t^4 \; mod \; n_t \\
&= H3(X)^{d_t}(Y(a_t^2+1)\beta^{-2})^{2d_t} Y^{-2d_t} H3(T_{ID})^{2d_t}(u_t^2+1)^{2d_t} b_t^4 \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}((a_t^2+1)(u_t^2+1))^{2d_t} \beta^{-4d_t} b_t^4 \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}(a_t^2u_t^2+a_t^2+u_t^2+1)^{2d_t} \beta^{-4d_t} \beta^{4d_t}(u_t-a_t)^{-4d_t} \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}((a_tu_t+1)^2+(u_t-a_t)^2)^{2d_t}(u_t-a_t)^{-4d_t} \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}((c^2+1)(u_t-a_t)^2)^{2d_t}(u_t-a_t)^{-4d_t} \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}((c^2+1)(u_t-a_t)^2)^{2d_t}(u_t-a_t)^{-4d_t} \; mod \; n_t \\
&= H3(X)^{d_t} H3(T_{ID})^{2d_t}(c^2+1)^{2d_t}
\end{aligned}
$$

The above reduction shows that the values $\{b_t, r_t, u_t\}$ that are learned from corresponding signature process record tuple $\{X, Y, a_t, \beta, t\}$ are eliminated. Hence, the Eq. (6.2) holds true for any values $\{b_t, r_t, u_t\}$ that are learned from other signature process records tuples as well. Therefore, the TPSP cannot link the token ID to a customer's corresponding signature process record using the unblinded signature.

6.3.2 Unforgeability

This section shows that adversaries cannot forge the TPSP's signature. First, we concentrate on a scenario where the attacker has no access to the valid signature (c, s). Anyone who wants to verify the TPSP's signature can prove it by performing the following computation:

$$s^{e_t} \equiv H3(X)H3(T_{ID})^2(c^2+1)^2 \; mod \; n_t$$

In order to successfully pass the signature verification without having full access to the signature, the attacker need to compute s in such way:

$$s \equiv H3(X)^{d_t} H3(T_{ID})^{2d_t}(c^2+1)^{2d_t} \; mod \; n_t$$

Even if the attacker has access to common information X, message T_{ID} and the part of signature c, it is computationally impossible to acquire the private component d_t through factorization of n_t. Additionally, even if the attacker has access to $s, H3(X)$ and $H3(T_{ID})$. it is still computationally impossible to compute c in such way $c^2 \equiv (s^{e_t} \cdot H3(X)^{-1} \cdot H3(T_{ID})^{-2})^{1/2} - 1 \; mod \; n_t$ without the factorization of n_t. Hence, the attacker cannot forge the signature in the first scenario.

Next we consider the scenario where the attacker has access to valid signature (c, s), common information X and the message T_{ID}. We prove that the attacker still cannot forge the signature and obtain another signature (c', s') on the a different Token ID T'_{ID} with the common information X'. Although the attacker has access to X and c, it cannot compute s' such that $s' \equiv s.H3(T_{ID})^{-2d_t} H3(T'_{ID})^{-2d_t} \; mod \; n_t$ without obtaining the value of d_t through factorization of n_t. Additionally it is also impossible to compute c' such that $c'^2 \equiv (s^{e_t} \cdot H3(X)^{-1} \cdot H3(T'_{ID})^{-2})^{1/2} - 1 \; mod \; n_t$ without factorization of n_t. Hence even if the attacker has access to a valid signature, it still cannot forge the TPSP's signature.

6.3.3 Exculpability

This section shows how our protocol prevents the merchant from double-spending prevention and reveals the identity of a dishonest customer who tries to double-spend the payment token.

Double Spending Prevention

We use a ZKP (Zero-Knowledge Proof) based authentication scheme so that the customer and the merchant can authenticate each other without any assistance from the TPSP. This authentication scheme also ensures that the merchant cannot double-spend the payment token. Whenever a user wants to spend a payment token, it has to prove that it is a registered user first. Such validation is carried through an exchange of challenge and response between a prover and a verifier. The challenge verifier issues vary for each transaction, and thus the same response cannot be repeated to prove the validity of a user. Hence, if the merchant tries to reuse the token with another merchant, it cannot create an appropriate response to a challenge issued by the other merchant as it has no access to a customer's secret identity key. Hence the payment attempt is refused.

Double Spending Detection

Assume that a customer aims to spend the payment token with two merchants. Only one merchant can successfully deposit the token as the TPSP stores the successfully deposited token IDs. Hence, the other merchant's deposit request is rejected. Even though no one is at a disadvantage here, there is no way to trace back the dishonest customer. We incorporate a tracing method to track the dishonest customer to tackle this problem. Customer generates a security tag T_c using R and public key P_k as follows:

$$T_c = P_k^{(1+R)}$$

At TPSP, if the T_{ID} already exists, it invokes the anonymity revocation protocol. Assume that the customer submits the same Token ID to merchants A and B. Merchant A has stored (T_{ID}, T_1, R_1) and merchant B has stored (T_{ID}, T_2, R_2) where T_1 and T_2 are security tags created using random numbers R_1 and R_2 respectively. TPSP can request merchants who have submitted the same Token ID to submit the security tags and the corresponding random numbers. It then can reveal the identity of the customer by calculating:

$$\begin{aligned}
&\left(\frac{T_2^{R_1}}{T_1^{R_2}}\right)^{(R_1-R_2)^{-1}} \\
&= \left(\frac{(P_k^{(1+R_2)})^{R_1}}{(P_k^{(1+R_1)})^{R_2}}\right)^{(R_1-R_2)^{-1}} \\
&= \left(P_k^{(R_1+R_1R_2-R_2-R_2R_1)}\right)^{(R_1-R_2)^{-1}} \\
&= \left(P_k^{(R_1-R_2)}\right)^{(R_1-R_2)^{-1}} \\
&= P_k
\end{aligned}$$

where P_k is the public key of the dishonest customer.

6.4 Experimental Evaluation for Security and Privacy Analysis

Formal analysis is a mathematically proven modelling approach to verify the security and privacy properties of a protocol. It uses symbolic and computational techniques for this purpose. We adopt the symbolic method to verify our proposed protocol. We model the stages and the security

and privacy properties of the protocol using the pi-calculus. We then verify whether our protocol satisfies these properties using the Proverif Automated verification tool. We verify the following security and privacy properties of our untraceable payment protocol using Proverif:

- Secrecy (Confidentiality) - This property ensures that a third-party adversary cannot acquire any information about the messages shared between the main actors of the protocol.
- Message Authenticity (Data origin authenticity) - This property ensures that the message is originated from an authorised source and not amended while in transit. We model the authenticity property according to the correspondence assertion of the events.
- Untraceability - The objective of our protocol is to hide the link between the token ID and the customer. We prove untraceability using the observational equivalence. We use *bisimilarity* to express equivalence which indicates that two processes are equivalent if an adversary cannot tell the difference between them.

6.4.1 Protocol Modelling in ProVerif Tool

Proverif is an automated tool used to verify the security and privacy properties of protocols (Blanchet et al., 2001). Initially, the developer created the software to verify a protocol's secrecy (reachability) properties. Then it is extended to support verification of authenticity (correspondence) (Blanchet, 2002) and privacy (equivalence) (Blanchet et al., 2008) properties. Proverif allows the user to model the protocol and properties in pi-calculus and translates them into prolog rules. Then it uses queries to test whether the protocol satisfies the modelled properties. If the properties are not satisfied, then the Proverif tool provides a trace of the attack. We refer to the manual for more details on the syntax used to model the protocol and the properties (Blanchet et al., 2018). The basic grammar of the terms and process behaviour used in pi-calculus are presented in Table 10.

The security and privacy properties are classified into two main classes: trace and equivalence. Trace property is defined and tested on each execution of the protocol. Example of such property is secrecy and authentication. These properties need to be satisfied every time the protocol runs. Equivalence properties are defined and tested as equivalences between concurrent executions of the protocol. If the two executions of the protocol are similar, then the attacker cannot distinguish

Table 10: Pi-Calculus Grammar Notations

Notation	Meaning
$M, N ::=$	Terms
a, b, c, k	names
x, y, z	variables
$(M_1, ...M_k)$	Tuple
$h(M_1, ...M_k)$	Constructor/Destructor
$M = N$	Term equality
$M <> N$	Term inequality
$P, Q, R ::=$	Processes
$in(M, x : t); P$	message input
$out(M, N); P$	message output
$let \quad x \quad = \quad M \; in \; P \; else \; Q$	term evaluation

between them. An example of such property is untraceability. Additionally, we use terms to denote the messages shared between actors, and cryptographic primitives such as hash, signature and encryption/decryption are represented using black boxes and considered perfectly secure.

6.4.2 Formalising the Protocol

This section presents the symbolic formal model of our untraceable payment protocol. Our protocol involves three actors: The customer, the merchant and the TPSP. We consider a Dolev-Yao (Dolev and Yao, 1983) kind of attacker who can eavesdrop, modify, inject and remove messages shared using public communication channels. However, this attacker can read or intercept messages only if it possesses the correct secret key. Hence, the presumption is that protocol restricts the attacker's ability under the perfect cryptography assumption. We also consider actors in our payment protocol to be either honest or fraudulent. Honest actors adhere to the protocol and carry out the assigned tasks. On the other hand, fraudulent or dishonest actors perform malicious actions such as double-spending and refuse to comply with the protocol.

In our modelling, we interpret the cryptographic primitives as symbolic functions. Symmetric encryption is represented by a constructor *senc*, which takes a *bitstring* and a *key* as input arguments and returns encrypted *bitstring* as output. A destructor *sdec* represents the symmetric decryption which takes an encrypted *bitstring* and a *key* as inputs and returns the message as an output. This can be formalised as below:

```
fun senc (bitstring, key): bitstring.
reduc forall m:bitstring, k:key; sdec(senc(m,k),k)= m.
```

We model the partially blind signature by combining our functional model with a well-known model used in Proverif for digital signatures. Similar to asymmetric encryption, digital signatures also rely on private (*sskey*) and public key (*spkey*). We use a constructor *spk* which takes *sskey* and returns *spkey* to denote generation of key-pairs. Digital signature signing is represented by a constructor *sign* which takes a *bitstring* and a *sskey* as inputs and returns a signature in the form of a *bitstring* as output. We use a destructor *checksign* to verify the signature. The destructor returns *true* if the verification is successful. This process is formalised as below:

```
fun spk (sskey): spkey.
fun sign (bitstring, sskey): bitstring.
reduc forall m:bitstring,k:sskey; checksign(sign(m,k),spk(k),m)= true.
```

The *blind* and *unblind* functions represent blinding a message and unblinding a blinded message to obtain the original message. This can be formalised as below:

```
fun blind (bitstring, bitstring): bitstring.
reduc forall m:bitstring,r:bitstring,k:sskey; unblind (sign(blind(m,r),k),r)= sign(m,k).
```

We model the protocol as a tuple of processes that represents the actors involved. We assume that the authentication stage is completed successfully between the customer and the merchant. We encrypt the communication between these actors using a symmetric key encryption algorithm. Hence, we assume the symmetric keys are shared after successfully executing the mutual authentication protocol.

Customer Process The actions undertaken by the customer are modelled using the customer process. The customer first sends an encrypted Purchase Request (*PR*) to the merchant and awaits for them to send a reply. After receiving the Purchase Request-Reply *PRR* from the merchant, the customer generates a Token ID (*TD*) and blinds the Token ID. The customer sends the encrypted blinded Token ID to the TPSP and then waits for the TPSP to reply with a blind signature. After receiving the blind signature, the customer sends an encrypted payment token and the unblinded signature to the merchant. Additionally we insert events such as *PurchaseRequest*, *TokenWithdrawal* and *PaymentbyToken* to indicate commencement of each phase of the protocol. We formalize the customer process as below:

```
let CUSTOMER(k1:key,k2:key,r:bitstring)=
event PurchaseRequest;
out(c,senc(PR,k1));
in (c,y:bitstring);
let PRR= sdec(y,k1) in
event TokenWithdrawal;
out(c,senc(blind (TD,r),k2));
in(c,x:bitstring);
let BS=sdec(x,k2) in
let sign=unblind(BS,r) in
event PaymentbyToken;
out(c,(senc(TD,k1),senc(sign,k1)));
```

Merchant Process The actions undertaken by the merchant are modelled using the merchant process. The merchant receives the Purchase Request (*PR*) from the customer and then sends a Purchase Request-Reply tuple *PRR* to the customer. After receiving the payment token from the customer, it first verifies the signature on the token. If the verification is successful, it sends a deposit request to the TPSP and awaits a reply. After TPSP sends the Deposit Confirmation *DC* message, the merchant sends the Purchase Confirmation *PC* message to the customer. Additionally we insert events such as *PurchaseRequest*, *PaymentbyToken* and *DepositRequest* to indicate commencement of each phase of the protocol. We formalise the merchant process as below:

```
let MERCHANT(k1:key,k3:key,pkS:spkey)=
event PurchaseRequest;
in (c,y:bitstring)
let PR = sdec(y,k1) in
out (c,senc(PRR,k1);
event PaymentbyToken;
in (c,(x:bitstring,z:bitstring));
let TD=sdec(x,k1) in
let sign=sdec(z,k1) in
let (=pkS,k:key) = checksign(sign,pkS) in
event DepositRequest;
out(c,(senc(TD,k3),senc(sign,k3)));
in(c,w:bitstring);
let DC = sdec(w,k3) in
out(c,senc(PC,k1) ;
```

TPSP The actions undertaken by the TPSP are modelled using the TPSP process. The TPSP issue a blind signature on the token and sends it to the customer. The customer then unblinds the signature and obtains the signature on the payment token. The customer then sends the token and the unblinded signature to the merchant. Subsequently, the merchant forwards the payment token and unblinded signature to the TPSP. We insert events such as *TokenWithdrawal* and *DepositRequest* to indicate the commencement of each phase of the protocol. We formalise the TPSP process as below:

```
let TPSP(k2:key,k3:key,skS:sskey)=
event TokenWithdrawal;
in (c,y:bitstring)
let TD = sdec(y,k2) in
out(c,senc(sign(TD,skS),k2));
event DepositRequest;
in(c,(x:bitstring,z:bitstring));
let TD = sdec(x,k3) in
let sign =sdec(z,k3) in
out(c,senc(DC,k3) ;
```

6.4.3 Protocol Evaluation

Secrecy

We prove confidentiality using reachability property in Proverif. We model the reachability property using a predicate query: $query\ attacker(s)$ where s is the corresponding message. This property allows us to investigate whether the attacker can access any messages communicated between the actors of the protocol. Hence we use it to evaluate the secrecy or confidentiality of the messages. We use Proverif to evaluate the messages sent and received by all the actors in our protocol using the public channel. Table 11 shows the final outcome of secrecy evaluation. A (✓) indicates that the protocol upholds the secrecy property, and ($\times$) indicates that the protocol does not uphold the property.

Table 11: Analysis of Secrecy Property

Message	Meaning	From	To	Result
PR	Purchase Request	Customer	Merchant	✓
PRR	Purchase Request Reply	Merchant	Customer	✓
TD	Token ID	Customer	TPSP	✓
BS	Blind Signature	TPSP	Customer	✓
DC	Deposit Confirmation	TPSP	Merchant	✓
PC	Purchase Confirmation	Merchant	Customer	✓

From Table 11 we can conclude that the attacker cannot capture any of the messages communicated between the customer, the merchant and the TPSP.

Authenticity

We verify the authenticity of the messages using the correspondence assertion property of Proverif. This property expresses a relationship between events. Assume a protocol execution involves executing two events e and e'. The correspondence property can indicate whether event e is executed after the event e'. We verify the authentication property of the proposed protocol using the correspondence assertion property as it expresses the relationship between the events in the form of "If an event is executed in the protocol, that means some other event has already been executed in the protocol".

In our protocol, we declare the following events:

- *PurchaseRequest* - This event indicates that the customer has authenticated themselves successfully with the merchant to commence the purchase.
- *TokenWithdrawal* - This event indicates that the customer has received a purchase reply from the merchant and can start the token withdrawal process with the TPSP. A customer can carry out this event if and only if it has successfully authenticated itself with the merchant. Hence the execution of *PurchaseRequest* event has to precede the execution of *TokenWithdrawal* event.
- *PaymentbyToken* - This event indicates that the TPSP has issued a valid signature on the token, and the customer can pay for the product using this valid payment token. A customer can carry out this event if and only if it has successfully obtained a valid payment token from the TPSP. Hence the execution of *TokenWithdrawal* event has to precede the execution of *PaymentbyToken* event.
- *DepositRequest* - This event indicates that the merchant has verified the token successfully and now can deposit the payment token with the TPSP. A merchant can carry out this event if it has successfully authenticated itself with the customer. Hence the execution of *PaymentbyToken* event has to precede the execution of *DepositRequest* event.

We consider the following correspondence assertions to prove authenticity :

- $inj - event(PurchaseRequest) \Rightarrow inj - event(TokenWithdrawal)$
- $inj - event(TokenWithdrawal) \Rightarrow inj - event(PaymentbyToken)$

- $inj - event(PaymentbyToken) \Rightarrow inj - event(DepositRequest)$

From Table 12, we can observe that the authenticity property of the protocol is upheld during each stage.

Table 12: Analysis of Authenticity Property

Event Correspondence	Result
PurchaseRequest-TokenWithdrawal	✓
TokenWithdrawal-PaymentbyToken	✓
PaymentbyToken-DepositRequest	✓

Untraceability

The untraceability property guarantees that the link between a customer and the Token ID remains hidden from the TPSP. So if the TPSP cannot trace the correct owner of a Token ID, it is not able to create a link between the merchant and the customer. We follow the following definition of untraceability and analyse our protocol in pi-calculus.

Definition 6. *For any two digital tokens $T1$ and $T2$ and for two customer processes $C1$ and $C2$ such that $T1$ and $T2$ are generated by $C1$ and $C2$ respectively, the untraceability property is upheld in our protocol if the TPSP cannot distinguish whether tokens T_i where $i = 1$ or 2 is generated by $C1$ or $C2$.*

We use equivalence property to prove the untraceability of our proposed payment protocol. We can say that our protocol satisfies the untraceability requirement if the customer process $C1$ with Token ID $T1$ paralleled by another customer process $C2$ with Token ID $T2$ is observationally equivalent to the customer process $C1$ with Token ID $T2$ paralleled by customer process $C2$ with Token ID $T1$. i.e $C1 \mid C2 \approx C1\{T2/T1\} \mid C2\{T1/T2\}$

Proverif confirms that our payment protocol upholds the untraceability property during each protocol execution. Because the payment token is signed using a partially blind signature, and thus, the TPSP cannot create any link between a Token ID and its generator during the deposit stage.

6.4.4 Computational Complexity Analysis

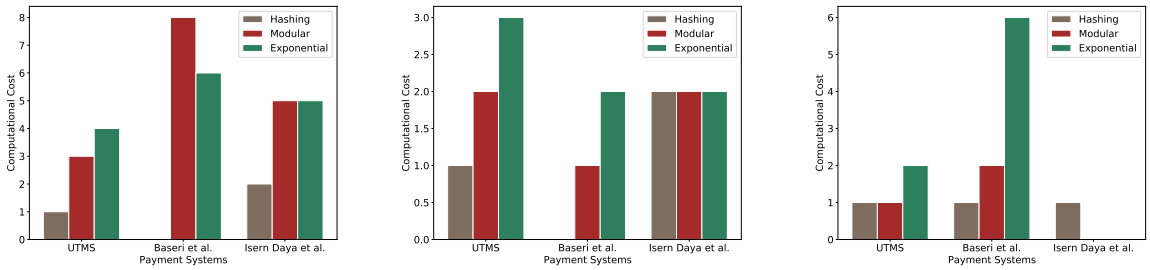
In Table 13, we compare the computational complexity of our proposed protocol with (Baseri et al., 2013) and (Deya et al., 2012). We use the following notations to indicate the computational

cost involved.

- H - indicates the computation time needed for a single hashing operation.
- M - indicates the computation time needed for a single modular operation.
- E - indicates the computation time needed for a single exponential operation.

Table 13: Computational Performance Analysis

Protocol Phase	UTMS	(Baseri et al., 2013)	(Deya et al., 2012)
Token Withdrawal (Customer)	1H+3M+4E	8M+6E	2H+5M+5E
Token Withdrawal (TPSP)	1H+2M+3E	1M+2E	2H+2M+2E
Token Deposit (Merchant)	1H+1M+2E	1H+2M+6E	1H



(a) Token Withdrawal (Customer) (b) Token Withdrawal (TPSP) (c) Token Deposit (Merchant)

Figure 29: Computational Cost Comparison

Fig 29a, 29b and 29c illustrate the computational cost comparison between our proposed untraceable mobile payment system (UTMS), (Baseri et al., 2013) and (Deya et al., 2012). Compared to the other two schemes, our system has lower computational costs on the customer side. The computational cost incurred at the TPSP is slightly higher in our system than the scheme proposed by (Baseri et al., 2013). But, the cost is comparatively lower compared to (Deya et al., 2012). At the token deposit phase, our system has a lower computational cost than (Baseri et al., 2013). The scheme proposed by (Deya et al., 2012) bears the lowest computational cost for the merchant at the token deposit stage. Because their scheme is an online scheme, the merchant does not have to perform any additional computations on their side apart from forwarding the deposit request to the TPSP. However, the drawback of their system is that it requires the TPSP to stay online during all the transactions to perform double-spending detection.

6.5 Conclusion

This chapter proposes an untraceable mobile payment system to prevent TPSPs from collecting additional details related to a transaction between a customer and a merchant. Our proposed system ensures security and privacy using cryptographic primitives such as blind signature, partially blind signature and zero-knowledge proof. The Proverif analysis shows that the proposed system is untraceable since the payment token cannot be linked to a customer. The evaluation also indicates that the proposed system preserves confidentiality and authenticity properties. Additionally, the proposed system grants customers the ability to request a refund if they are unhappy with the product. Finally, the system assures that customers and merchants can't commit double-spending frauds or forge the TPSP's signature. Meanwhile, it also preserves efficiency and low computational cost on the customer side as it does not use cryptographic primitives that incur a higher computational cost.

Chapter 7

Conclusion

This thesis introduces various approaches to preserving privacy in state-of-the-art recommendation systems and third-party based payment systems. We now provide an overview of the contributions that have been made and recommend future research directions that can further contribute to the existing body of literature. We also discuss the limitations of our work in this section.

7.1 Overview of Contributions

We design effective, efficient, and privacy-preserving recommendation and payment system frameworks by employing cryptographic primitives, local differential privacy, and federated learning technologies.

First, we address our first objective, "To design an effective input perturbation mechanism for CF-based recommendation systems that protects the user's privacy from the TPSP while offering optimal utility to the TPSP". We investigate the challenge of ensuring users' privacy without compromising the utility for the TPSP in recommendation systems. In particular, we consider a MF-based recommendation system which uses user ratings as input. We propose the Bounded Laplace mechanism (BLP) as an input perturbation mechanism on the user side that regulates the effect of noise added to the user ratings by confining the perturbed ratings to a predetermined domain. Through BLP, we ensure that utility is not sacrificed to provide privacy protection. Furthermore, we also introduce Matrix Factorization with a Mixture of Gaussian (MF-MoG) as a noise estimation and rating prediction component at the TPSP's side to further improve the util-

ity. We demonstrate through empirical evaluation that the BLP and noise estimation component plays a vital role in enhancing the utility of a matrix factorization-based recommendation system while offering strong privacy protection to the users. The empirical evaluations also show that our method outperforms the existing LDP-based recommendation models such as (Shin et al., 2018) and (Berlioz et al., 2015). Additionally, our approach causes much lower communication costs compared to existing LDP-based recommendation systems, e.g.(Shin et al., 2018).

Second, we extend the above privacy-preserving recommendation framework to a hybrid recommendation system in an attempt to achieve the second objective, "To design a privacy-preserving hybrid CF-based recommendation that uses ratings and reviews as input while offering optimal utility to the TPSP". We use a hybrid recommendation system which integrates a deep learning-based sentiment analysis model, BERT, with a MF-based recommendation system. To preserve privacy in such systems, we use BLP to perturb the user ratings and BERT to tokenize the reviews locally before sending them to the TPSP. The experiments conducted with two amazon review and rating datasets demonstrate that the utility of our proposed privacy-preserving hybrid recommendation system (BERT-MF-MoG) outperforms the MF-MoG recommendation system while offering the same level of privacy protection to the users.

Third, we focus on applying BLP in federated matrix factorization for implicit feedback. In particular, this addresses the objective, "To design a privacy-preserving federated matrix factorization based recommendation system which can hide the user-item interaction data from the TPSP". Here the focus is on using BLP as an iterative input perturbation mechanism which prevents stochastic gradients from revealing any sensitive information to the TPSP. We provide a sufficient condition for BLP to satisfy ε-differential privacy when perturbing gradients locally at the user-side. On the one hand, we ensure that user-item interaction is hidden from the TPSP by introducing a federated learning-based collaborative filtering system. On the other hand, stochastic gradient perturbation at the user-side ensures that the TPSP cannot infer any sensitive information. We have empirically evaluated the privacy-utility trade-off in our system using Movielens and Jester Dataset. Our empirical evaluations demonstrate that the utility of our recommendation system relies on user/item set size, the number of iterations used for training and the privacy budget ε.

Finally, we focus on the last objective, "To design a secure and privacy-preserving TPSP-based payment system so that the TPSP will not be able to track any transaction-related information". We

answer this question by proposing an untraceable TPSP-based payment system built using cryptographic primitives such as partially blind signature and zero-knowledge proof. We use Proverif to verify the untraceability, confidentiality and authenticity of our proposed payment system. We also ensure that customers and merchants couldn't execute double-spending attacks on our payment system. The proposed approach causes low computational costs on the customer side as it does not use cryptographic primitives that require complex computations.

7.2 Limitations

The work presented in this thesis does have certain limitations, which are listed below:

- If the recommendation models were made public, they would be susceptible to reconstruction attacks, model inversion attacks, and membership inference attacks. Our proposed models do not account for such attacks since the data is perturbed and transferred to the TPSP. Hence, the original data is not revealed to an adversary. While beyond the scope of this thesis, there are methods available to mitigate such attacks to further strengthen the system's security. For example, employing a differential privacy algorithm alongside cryptographic primitives such as secure multiparty computation or a proxy network hides the linkage between the perturbed data and the users.

- The untraceable third-party payment system uses cryptographic primitives such as zero-knowledge proof and partially blind signatures. Even though we formally verified the security and privacy properties of the protocol using Proverif, the efficiency of the protocol is evaluated based on counting hashing, modular and exponential operations. Real-world prototype implementation and performance comparison will further prove the suitability of the protocol in e-commerce environments.

- In the proposed LDP-based recommendation systems the privacy budget is universal and decided by the TPSP. Hence, the same level of privacy is offered to all users. However, it is very common for users to have different expectations and requirements regarding the privacy level that is suitable to protect their data. As a result of using a global privacy budget, the privacy-enhanced recommendation systems might provide insufficient privacy protection to some users while over-protecting others. Additionally, by adopting personalised differential

privacy definition in our models we should be able to attain a higher level of utility when we do not provide strong privacy protection to users who do not require it.

- The hybrid recommendation models protects the privacy of users by processing the reviews and adding noise to the user's actual ratings locally. Since the reviews are processed locally and only the word vectors are sent for aggregation, we assume the TPSP cannot infer actual sentiment of the user from the word vectors. However, the TPSP still can infer the sentiment of the user by converting the word vectors to actual words. Even though we could not find a well-known algorithm to convert vectors to words, the TPSP can achieve this by simply designing a distance function. For example, if the TPSP receives a word vector v they can interpret this vector as a word by finding a word with a similar word vector v_w such that $d(v, v_w)$ is small. Here d is the distance function which measures how similar both vectors are.

7.3 Future Works

The work in this thesis concentrates mainly on achieving an ideal privacy-utility trade-off in recommendation and payment systems. Though we propose privacy-preserving practices for both scenarios in this thesis, the problem is still far from being solved as the state-of-the-art technologies continue to evolve, allowing TPSPs to collect various types of data from the users. Our future work will follow the same direction that we have taken in this thesis, as privacy and utility trade-offs are essential in accomplishing an ideal solution to this problem.

- **Reducing privacy cost in federated learning models:** The training has to go through several iterations in federated learning models before reaching convergence. Due to its sequential composition property in differential privacy-based solutions, the privacy cost keeps rising with the number of iterations. Rising privacy cost is a concern in state-of-the-art recommendation systems as a higher value for privacy budget ε means that privacy loss incurred is also high. Reducing the number of iterations might resolve this issue. However, such an approach will affect the utility as the model might not have achieved its optimal convergence point due to fewer iterations. Therefore, exploring whether the number of iterations can be significantly reduced without causing much privacy loss will be an exciting avenue of re-

search. Note that such an approach will differ from what is practised by the state-of-the-art recommendation systems. This approach should ensure that the training process of the new recommendation model has to converge with fewer training iterations. We can explore the usefulness of meta-learning and neural networks as suitable solutions to this problem.

- **Personalised Differential Privacy:** Most TPSPs are inclined to aggregate user data, such as location history, browsing history, views, likes etc., to provide a high-quality personalised experience for the customers. Even though we considered all the information sensitive in our thesis, some information might not reveal any sensitive information about the users. Hence, perturbing all the user data does not yield optimal utility for the recommendation systems. To address this issue, we can concentrate on introducing personalised user privacy based concept where the user decides which kind of information is sensitive to them and what level of privacy they prefer. In such cases, the recommendation model can learn the actual preferences of the user, which in turn will increase the utility of the system while providing higher privacy protection to those who opt for it. The key idea is to identify the sensitive item categories for each user and the contribution of these items to privacy loss.

- **Privacy loss in context-aware recommendation systems:** Recommendation models such as matrix factorization fall short when different inputs are used to estimate a user's preferences. As a result, TPSPs turn their attention toward deep learning recommendation models. Simply adopting existing privacy technologies for such complicated models will not yield an optimal privacy-utility trade-off. Hence introducing new privacy-preserving data aggregation and analytical practices can be a direction that needs more attention.

Acronyms

ALS Alternating Least Squares 12

BERT Bidirectional Encoder Representations from Transformers 63

CB Content-based Filtering 9

CF Collaborative Filtering 2

CNN Convolutional Neural Networks 62

DP Differential Privacy 3

EM Expectation Maximization 50

HS Hybrid Systems 9

ISGD Input Perturbation Method 55

KNN K-Nearest Neighbour 10

LDP Local Differential Privacy 3

MF Matrix Factorization 2

PG-MF Private Gradient-Matrix Factorization 56

RMSE Root Mean Squared Error 53

SGD Stochastic Gradient Descent 12

TPSP Third-Party Service Provider 2

ZKP Zero-Knowledge Proof 23

References

Abe, M. and Fujisaki, E. (1996), How to date blind signatures, *in* 'International Conference on the Theory and Application of Cryptology and Information Security', Springer, pp. 244–251.

Adomavicius, G. and Tuzhilin, A. (2005), 'Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *IEEE transactions on knowledge and data engineering* **17**(6), 734–749.

Aggarwal, C. C. (2016), Model-based collaborative filtering, *in* 'Recommender systems', Springer, pp. 71–138.

Ammad-Ud-Din, M., Ivannikova, E., Khan, S. A., Oyomno, W., Fu, Q., Tan, K. E. and Flanagan, A. (2019), 'Federated collaborative filtering for privacy-preserving personalized recommendation system', *arXiv preprint arXiv:1901.09888* .

Anderson, R., Manifavas, C. and Sutherland, C. (1996), Netcard—a practical electronic-cash system, *in* 'International Workshop on Security Protocols', Springer, pp. 49–57.

Angwin, J., Larson, J., Mattu, S. and Kirchner, L. (2016), Machine bias, *in* 'Ethics of Data and Analytics', Auerbach Publications, pp. 254–264.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D. and Shmatikov, V. (2020), How to backdoor federated learning, *in* 'International Conference on Artificial Intelligence and Statistics', PMLR, pp. 2938–2948.

Baldimtsi, F. and Lysyanskaya, A. (2013), Anonymous credentials light, *in* 'Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security', pp. 1087–1098.

Banerjee, S., Hegde, N. and Massoulié, L. (2012), The price of privacy in untrusted recommendation engines, *in* '2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)', IEEE, pp. 920–927.

Baseri, Y., Takhtaei, B. and Mohajeri, J. (2013), 'Secure untraceable off-line electronic cash system', *Scientia Iranica* **20**(3), 637–646.

Bassily, R., Nissim, K., Stemmer, U. and Guha Thakurta, A. (2017), 'Practical locally private heavy hitters', *Advances in Neural Information Processing Systems* **30**.

Bennett, J., Lanning, S. et al. (2007), The netflix prize, *in* 'Proceedings of KDD cup and workshop', Vol. 2007, Citeseer, p. 35.

Berlioz, A., Friedman, A., Kaafar, M. A., Boreli, R. and Berkovsky, S. (2015), Applying differential privacy to matrix factorization, *in* 'Proceedings of the 9th ACM Conference on Recommender Systems', pp. 107–114.

Bhagoji, A. N., Chakraborty, S., Mittal, P. and Calo, S. (2019), Analyzing federated learning through an adversarial lens, *in* 'International Conference on Machine Learning', PMLR, pp. 634–643.

Bilge, A., Gunes, I. and Polat, H. (2014), 'Robustness analysis of privacy-preserving model-based recommendation schemes', *Expert Systems with Applications* **41**(8), 3671–3681.

Blanchet, B. (2002), From secrecy to authenticity in security protocols, *in* 'International Static Analysis Symposium', Springer, pp. 342–359.

Blanchet, B., Abadi, M. and Fournet, C. (2008), 'Automated verification of selected equivalences for security protocols', *The Journal of Logic and Algebraic Programming* **75**(1), 3–51.

Blanchet, B., Smyth, B., Cheval, V. and Sylvestre, M. (2018), 'Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial', *Version from* pp. 05–16.

Blanchet, B. et al. (2001), An efficient cryptographic protocol verifier based on prolog rules., *in* 'csfw', Vol. 1, Citeseer, pp. 82–96.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A. and Seth, K. (2017), Practical secure aggregation for privacy-preserving machine learning, *in* 'proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security', pp. 1175–1191.

Brozovsky, L. and Petricek, V. (2007), 'Recommender system for online dating service', *arXiv preprint cs/0703042* .

Buchanan, A. M. and Fitzgibbon, A. W. (2005), Damped newton algorithms for matrix factorization with missing data, *in* '2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)', Vol. 2, IEEE, pp. 316–322.

Cadwalladr, C. and Graham-Harrison, E. (2018), 'How cambridge analytica turned facebook 'likes' into a lucrative political tool', *The Guardian* **18**.

Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W. and Shmatikov, V. (2011), " you might also like:" privacy risks of collaborative filtering, *in* '2011 IEEE symposium on security and privacy', IEEE, pp. 231–246.

Cao, T., Lin, D. and Xue, R. (2005), 'A randomized rsa-based partially blind signature scheme for electronic cash', *Computers & Security* **24**(1), 44–49.

Cen, Y., Zhang, J., Wang, G., Qian, Y., Meng, C., Dai, Z., Yang, H. and Tang, J. (2019), 'Trust relationship prediction in alibaba e-commerce platform', *IEEE Transactions on Knowledge and Data Engineering* **32**(5), 1024–1035.

Chai, D., Wang, L., Chen, K. and Yang, Q. (2020), 'Secure federated matrix factorization', *IEEE Intelligent Systems* **36**(5), 11–20.

Chaum, D. (1983), Blind signatures for untraceable payments, *in* 'Advances in cryptology', Springer, pp. 199–203.

Chaum, D., Fiat, A. and Naor, M. (1988), Untraceable electronic cash, *in* 'Conference on the Theory and Application of Cryptography', Springer, pp. 319–327.

Chen, F., Luo, M., Dong, Z., Li, Z. and He, X. (2018), 'Federated meta-learning with fast convergence and efficient communication', *arXiv preprint arXiv:1802.07876* .

Cheng, Z., Ding, Y., Zhu, L. and Kankanhalli, M. (2018), Aspect-aware latent factor model: Rating prediction with ratings and reviews, *in* 'Proceedings of the 2018 world wide web conference', pp. 639–648.

Chien, H.-Y., Jan, J.-K. and Tseng, Y.-M. (2001), Rsa-based partially blind signature with low computation, *in* 'Proceedings. Eighth International Conference on Parallel and Distributed Systems. ICPADS 2001', IEEE, pp. 385–389.

De La Torre, F. and Black, M. J. (2003), 'A framework for robust subspace learning', *International Journal of Computer Vision* **54**(1), 117–142.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1), 1–22.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Deya, A. P. I., Rotger, L. H., Capella, M. M. P. and Puigserver, M. M. (2012), 'Anonymous, fair and untraceable micropayment scheme: Application to lbs', *IEEE Latin America Transactions* **10**(3), 1774–1784.

Dolev, D. and Yao, A. (1983), 'On the security of public key protocols', *IEEE Transactions on information theory* **29**(2), 198–208.

Du, Y., Zhou, D., Xie, Y., Shi, J. and Gong, M. (2021), 'Federated matrix factorization for privacy-preserving recommender systems', *Applied Soft Computing* **111**, 107700.

Duchi, J. C., Jordan, M. I. and Wainwright, M. J. (2013), Local privacy and statistical minimax rates, *in* '2013 IEEE 54th Annual Symposium on Foundations of Computer Science', IEEE, pp. 429–438.

Duhigg, C. (2012), 'How companies learn your secrets', *The New York Times* **16**(2), 1–16.

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

Erlingsson, Ú., Pihur, V. and Korolova, A. (2014), Rappor: Randomized aggregatable privacy-preserving ordinal response, *in* 'Proceedings of the 2014 ACM SIGSAC conference on computer and communications security', pp. 1054–1067.

Erway, C. C., Küpçü, A., Hinkle, T. and Lysyanskaya, A. (2010), {ZKPDL}: A {Language-Based} system for efficient {Zero-Knowledge} proofs and electronic cash, *in* '19th USENIX Security Symposium (USENIX Security 10)'.

Fan, C.-I. and Lei, C.-L. (2002), 'A user efficient fair blind signature scheme for untraceable electronic cash', *Journal of Information Science and Engineering* **18**(1), 47–58.

Ferguson, N. (1993), Single term off-line coins, *in* 'EUROCRYPT'.

Foley, S., Karlsen, J. R. and Putniņš, T. J. (2019), 'Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies?', *The Review of Financial Studies* **32**(5), 1798–1853.

Foner, L. N. (1999), Political artifacts and personal privacy: The Yenta multi-agent distributed matchmaking system, PhD thesis, Massachusetts Institute of Technology.

Friedman, A. and Schuster, A. (2010), Data mining with differential privacy, *in* 'Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 493–502.

Geiping, J., Bauermeister, H., Dröge, H. and Moeller, M. (2020), 'Inverting gradients-how easy is it to break privacy in federated learning?', *Advances in Neural Information Processing Systems* **33**, 16937–16947.

Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. (2001), 'Eigentaste: A constant time collaborative filtering algorithm', *information retrieval* **4**(2), 133–151.

Google, S. (2019), 'How messages improves suggestions with federated technology - messages help'.
URL: https://support.google.com/messages/answer/9327902

Harper, F. M. and Konstan, J. A. (2015), 'The movielens datasets: History and context', *Acm transactions on interactive intelligent systems (tiis)* **5**(4), 1–19.

Hastie, T., Mazumder, R., Lee, J. D. and Zadeh, R. (2015), 'Matrix completion and low-rank svd via fast alternating least squares', *The Journal of Machine Learning Research* **16**(1), 3367–3402.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.-S. (2017), Neural collaborative filtering, *in* 'Proceedings of the 26th international conference on world wide web', pp. 173–182.

He, X., Zhang, H., Kan, M.-Y. and Chua, T.-S. (2016), Fast matrix factorization for online recommendation with implicit feedback, *in* 'Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval', pp. 549–558.

Heilman, E., Kendler, A., Zohar, A. and Goldberg, S. (2015), Eclipse attacks on bitcoin's peer-to-peer network, *in* '24th USENIX Security Symposium (USENIX Security 15)', pp. 129–144.

Holohan, N., Antonatos, S., Braghin, S. and Mac Aonghusa, P. (2020), 'The bounded laplace mechanism in differential privacy', *Journal of Privacy and Confidentiality* **10**(1).

Hu, Y., Koren, Y. and Volinsky, C. (2008), Collaborative filtering for implicit feedback datasets, *in* '2008 Eighth IEEE international conference on data mining', Ieee, pp. 263–272.

Hua, J., Xia, C. and Zhong, S. (2015), *Twenty-Fourth International Joint Conference on Artificial Intelligence*, p. 1763–1770.

Huo, Z., Yang, Q., Gu, B., Huang, L. C. et al. (2020), 'Faster on-device training using new federated momentum algorithm', *arXiv preprint arXiv:2002.02090* .

Jain, Y. K. and Bhandare, S. K. (2011), 'Min max normalization based data perturbation method for privacy protection', *International Journal of Computer & Communication Technology* **2**(8), 45–50.

Jeckmans, A. J., Beye, M., Erkin, Z., Hartel, P., Lagendijk, R. L. and Tang, Q. (2013), Privacy in recommender systems, *in* 'Social media retrieval', Springer, pp. 263–281.

Jiang, P. and Ying, L. (2020), An optimal stopping approach for iterative training in federated learning, *in* '2020 54th Annual Conference on Information Sciences and Systems (CISS)', IEEE, pp. 1–6.

Karame, G. O., Androulaki, E. and Capkun, S. (2012), Double-spending fast payments in bitcoin, *in* 'Proceedings of the 2012 ACM conference on Computer and communications security', pp. 906–917.

Kendall, A., Grimes, M. and Cipolla, R. (2015), Posenet: A convolutional network for real-time 6-dof camera relocalization, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 2938–2946.

Koren, Y. (2008), Factorization meets the neighborhood: a multifaceted collaborative filtering model, *in* 'Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 426–434.

Koren, Y., Bell, R. and Volinsky, C. (2009), 'Matrix factorization techniques for recommender systems', *Computer* **42**(8), 30–37.

Kosinski, M., Stillwell, D. and Graepel, T. (2013), 'Private traits and attributes are predictable from digital records of human behavior', *Proceedings of the national academy of sciences* **110**(15), 5802–5805.

Kumar, M., Katti, C. P. and Saxena, P. C. (2017), A secure anonymous e-voting system using identity-based blind signature scheme, *in* 'International conference on information systems security', Springer, pp. 29–49.

Lam, S. K., Frankowski, D., Riedl, J. et al. (2006), Do you trust your recommendations? an exploration of security and privacy issues in recommender systems, *in* 'International conference on emerging trends in information and communication security', Springer, pp. 14–29.

Liu, Z., Wang, Y.-X. and Smola, A. (2015), Fast differentially private matrix factorization, *in* 'Proceedings of the 9th ACM Conference on Recommender Systems', pp. 171–178.

Luo, M., Chen, F., Cheng, P., Dong, Z., He, X., Feng, J. and Li, Z. (2020), Metaselector: Meta-learning for recommendation with user-level adaptive model selection, *in* 'Proceedings of The Web Conference 2020', pp. 2507–2513.

Lysyanskaya, A. and Ramzan, Z. (1998), Group blind digital signatures: A scalable solution to electronic cash, *in* 'International Conference on Financial Cryptography', Springer, pp. 184–197.

Ma, L., Ge, Y. and Zhu, Y. (2014), 'Tinyzkp: a lightweight authentication scheme based on zero-knowledge proof for wireless body area networks', *Wireless personal communications* **77**(2), 1077–1090.

Martínez-Peláez, R. and Rico-Novella, F. J. (2006), New electronic cash model: a script anonym, *in* 'Proc. of the IADIS International Conference on E-Commerce,(e-commerce'06)', pp. 392–396.

McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B. A. (2017), Communication-efficient learning of deep networks from decentralized data, *in* 'Artificial intelligence and statistics', PMLR, pp. 1273–1282.

McSherry, F. and Mironov, I. (2009), Differentially private recommender systems: Building privacy into the netflix prize contenders, *in* 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 627–636.

Meng, D. and De La Torre, F. (2013), Robust matrix factorization with unknown noise, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 1337–1344.

Meng, X., Wang, S., Shu, K., Li, J., Chen, B., Liu, H. and Zhang, Y. (2018), Personalized privacy-preserving social recommendation, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 32.

Mobasher, B., Burke, R., Bhaumik, R. and Williams, C. (2007), 'Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness', *ACM Transactions on Internet Technology (TOIT)* **7**(4), 23–es.

Nakamoto, S. (2008), 'Bitcoin: A peer-to-peer electronic cash system', *Decentralized Business Review* p. 21260.

Narayanan, A. and Shmatikov, V. (2008), Robust de-anonymization of large sparse datasets, *in* '2008 IEEE Symposium on Security and Privacy (sp 2008)', IEEE, pp. 111–125.

Ni, J., Li, J. and McAuley, J. (2019), Justifying recommendations using distantly-labeled reviews and fine-grained aspects, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)', pp. 188–197.

Nishio, T. and Yonetani, R. (2019), Client selection for federated learning with heterogeneous resources in mobile edge, *in* 'ICC 2019-2019 IEEE international conference on communications (ICC)', IEEE, pp. 1–7.

Osman, N., Noah, S. and Darwich, M. (2019), 'Contextual sentiment based recommender system to provide recommendation in the electronic products domain', *International Journal of Machine Learning and Computing* **9**(4), 425–431.

Pal, A., Parhi, P. and Aggarwal, M. (2017), An improved content based collaborative filtering algorithm for movie recommendations, *in* '2017 tenth international conference on contemporary computing (IC3)', IEEE, pp. 1–3.

Pan, N., Yao, W. and Li, X. (2021), Friends recommendation based on kbert-cnn text classification model, *in* '2021 International Joint Conference on Neural Networks (IJCNN)', IEEE, pp. 1–6.

Papagelis, M. and Plexousakis, D. (2005), 'Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents', *Engineering Applications of Artificial Intelligence* **18**(7), 781–789.

Parameswaran, R. and Blough, D. M. (2008), 'Privacy preserving data obfuscation for inherently clustered data', *International Journal of Information and Computer Security* **2**(1), 4–26.

Paterek, A. (2007), Improving regularized singular value decomposition for collaborative filtering, *in* 'Proceedings of KDD cup and workshop', Vol. 2007, pp. 5–8.

Pathak, R. and Wainwright, M. J. (2020), 'Fedsplit: An algorithmic framework for fast federated optimization', *Advances in Neural Information Processing Systems* **33**, 7057–7066.

Preethi, G., Krishna, P. V., Obaidat, M. S., Saritha, V. and Yenduri, S. (2017), Application of deep learning to sentiment analysis for recommender system on cloud, *in* '2017 International conference on computer, information and telecommunication systems (CITS)', IEEE, pp. 93–97.

Preibusch, S., Peetz, T., Acar, G. and Berendt, B. (2016), 'Shopping for privacy: Purchase details leaked to paypal', *Electronic Commerce Research and Applications* **15**, 52–64.

Qin, Z., Yang, Y., Yu, T., Khalil, I., Xiao, X. and Ren, K. (2016), Heavy hitter estimation over set-valued data with local differential privacy, *in* 'Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security', pp. 192–203.

Raghavan, S., Gunasekar, S. and Ghosh, J. (2012), Review quality aware collaborative filtering, *in* 'Proceedings of the sixth ACM conference on Recommender systems', pp. 123–130.

Reid, F. and Harrigan, M. (2013), An analysis of anonymity in the bitcoin system, *in* 'Security and privacy in social networks', Springer, pp. 197–223.

Reuters (2018), 'China's cyber watchdog scolds ant financial over user privacy breach'. URL: https://www.reuters.com/article/us-ant-financial-china-idUKKBN1F006B

Sai Anand, R. and Madhavan, C. (2000), An online, transferable e-cash payment system, *in* 'International Conference on Cryptology in India', Springer, pp. 93–103.

Salinca, A. (2017), 'Convolutional neural networks for sentiment classification on business reviews', *arXiv preprint arXiv:1710.05978* .

San, A. M. and Sathitwiriyawong, C. (2016), Efficient offline micropayment protocol for multivendor, *in* '2016 International Computer Science and Engineering Conference (ICSEC)', IEEE, pp. 1–4.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001), Item-based collaborative filtering recommendation algorithms, *in* 'Proceedings of the 10th international conference on World Wide Web', pp. 285–295.

Sattler, F., Wiedemann, S., Müller, K.-R. and Samek, W. (2019), 'Robust and communication-efficient federated learning from non-iid data', *IEEE transactions on neural networks and learning systems* **31**(9), 3400–3413.

Shaowen, L. and Yong, C. (2017), An improved collaborative filtering recommendation algorithm, *in* '2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)', IEEE, pp. 204–208.

Shen, Y. and Jin, H. (2014), Privacy-preserving personalized recommendation: An instance-based

approach via differential privacy, *in* '2014 IEEE International Conference on Data Mining', IEEE, pp. 540–549.

Shin, H., Kim, S., Shin, J. and Xiao, X. (2018), 'Privacy enhanced matrix factorization for recommendation with local differential privacy', *IEEE Transactions on Knowledge and Data Engineering* **30**(9), 1770–1782.

Singel, R. (2009), 'Netflix spilled your brokeback mountain secret, lawsuit claims', *Threat Level (blog), Wired* .

Smith, B. and Linden, G. (2017), 'Two decades of recommender systems at amazon.com', *Ieee internet computing* **21**(3), 12–18.

Smith, V., Chiang, C.-K., Sanjabi, M. and Talwalkar, A. S. (2017), 'Federated multi-task learning', *Advances in neural information processing systems* **30**.

Spiekermann, S. and Cranor, L. F. (2008), 'Engineering privacy', *IEEE Transactions on software engineering* **35**(1), 67–82.

Srebro, N. and Jaakkola, T. (2003), Weighted low-rank approximations, *in* 'Proceedings of the 20th international conference on machine learning (ICML-03)', pp. 720–727.

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

Turk, M. A. and Pentland, A. P. (1991), Face recognition using eigenfaces, *in* 'Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition', IEEE Computer Society, pp. 586–587.

Vakeel, K. A., Das, S., Udo, G. J. and Bagchi, K. (2017), 'Do security and privacy policies in b2b and b2c e-commerce differ? a comparative study using content analysis', *Behaviour & Information Technology* **36**(4), 390–403.

Van den Oord, A., Dieleman, S. and Schrauwen, B. (2013), 'Deep content-based music recommendation', *Advances in neural information processing systems* **26**.

Vidal, R., Tron, R. and Hartley, R. (2008), 'Multiframe motion segmentation with missing data using powerfactorization and gpca', *International Journal of Computer Vision* **79**(1), 85–105.

Wang, Q. (2011), 'Compact k-spendable e-cash with anonymity control based offline ttp', *International Journal of Innovative Computing, Information and Control* **7**(1), 459–469.

Wang, T., Blocki, J., Li, N. and Jha, S. (2017), Locally differentially private protocols for frequency estimation, *in* '26th USENIX Security Symposium (USENIX Security 17)', pp. 729–745.

Wang, Y., Wang, M. and Xu, W. (2018), 'A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework', *Wireless Communications and Mobile Computing* **2018**.

Wei, W., Liu, L., Loper, M., Chow, K.-H., Gursoy, M. E., Truex, S. and Wu, Y. (2020), A framework for evaluating client privacy leakages in federated learning, *in* 'European Symposium on Research in Computer Security', Springer, pp. 545–566.

Weinsberg, U., Bhagat, S., Ioannidis, S. and Taft, N. (2012), Blurme: Inferring and obfuscating user gender based on ratings, *in* 'Proceedings of the sixth ACM conference on Recommender systems', pp. 195–202.

Xiong, X., Liu, S., Li, D., Cai, Z. and Niu, X. (2020), 'A comprehensive survey on local differential privacy', *Security and Communication Networks* .

Yakut, I. and Polat, H. (2010), 'Privacy-preserving svd-based collaborative filtering on partitioned data', *International Journal of Information Technology & Decision Making* **9**(03), 473–502.

Yang, E., Huang, Y., Liang, F., Pan, W. and Ming, Z. (2021), 'Fcmf: Federated collective matrix factorization for heterogeneous collaborative filtering', *Knowledge-Based Systems* **220**, 106946.

Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D. and Beaufays, F. (2018), 'Applied federated learning: Improving google keyboard query suggestions', *arXiv preprint arXiv:1812.02903* .

Zhang, F. and Kim, K. (2003), Efficient id-based blind signature and proxy signature from bilinear pairings, *in* 'Australasian Conference on Information Security and Privacy', Springer, pp. 312–323.

Zhang, J., Huo, L., Liu, X., Sui, C., Li, Z. and Ma, J. (2015), Transferable optimal-size fair e-cash with optimal anonymity, *in* '2015 International Symposium on Theoretical Aspects of Software Engineering', IEEE, pp. 139–142.

Zhang, L., Zhang, F., Qin, B. and Liu, S. (2011), 'Provably-secure electronic cash based on certificateless partially-blind signatures', *Electronic Commerce Research and Applications* **10**(5), 545–552.

Zhang, Z., Wang, T., Li, N., He, S. and Chen, J. (2018), Calm: Consistent adaptive local marginal for marginal release under local differential privacy, *in* 'Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security', pp. 212–229.

Zhao, X., Lv, Y. and He, W. (2009), A novel micropayment scheme with complete anonymity, *in* '2009 Fifth International Conference on Information Assurance and Security', Vol. 1, IEEE, pp. 638–642.

Zhu, L., Liu, Z. and Han, S. (2019), 'Deep leakage from gradients', *Advances in Neural Information Processing Systems* **32**.