Data Privacy in Supply Chains and Machine Learning through Differential Privacy and Cryptography

Li, T.

DOI
10.4233/uuid:56ad5e2c-c5ec-4a5e-ac0e-643474fb23c2

Publication date
2024

Document Version
Final published version

Citation (APA)
Li, T. (2024). *Data Privacy in Supply Chains and Machine Learning through Differential Privacy and Cryptography*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:56ad5e2c-c5ec-4a5e-ac0e-643474fb23c2

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Data Privacy in Supply Chains and Machine Learning through Differential Privacy and Cryptography

Data Privacy in Supply Chains and Machine Learning through Differential Privacy and Cryptography

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Friday 20 December 2024 at 10:00 o'clock

by

Tianyu Li

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. ir. R.L. Lagendijk	Delft University of Technology, *promotor*
Dr. Z. Erkin	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. M.M. de Weerdt	Delft University of Technology
Prof. dr. G. Smaragdakis	Delft University of Technology
Prof. dr. M.E. van Dijk	Centrum Wiskunde & Informatica (CWI), Netherlands
Prof. dr. M. Önen	Eurecom, France
Prof. dr. S. Roos	University of Kaiserslautern-Landau, Germany

Keywords: Data Privacy, Differential Privacy, Applied Cryptography, Machine Learning

Printed by: ProefschriftMaken Printing

Front & Back: Tingting and Tianyu Li

ISBN 978-94-6510-362-4

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Love is the meaning of life.

CONTENTS

SUMMARY

In modern society, data is increasingly important for people's daily lives and commercial activities to benefit organization, management and analysis. Proper data utilization involves different stages, and the data life cycle is introduced to describe the procedures as generation, collection, processing, storage, management, analysis, visualization and interpretation. Along the data life cycle, data privacy is highly concerned since possible breaches can lead to the abuse of personal information and financial loss. In this thesis, we advance data privacy protection in data processing, data management, and data analysis correlated with the *Spark Living Lab* project in the domain of supply chains and machine learning. To enhance privacy protection, we propose solutions based on differential privacy and cryptographic protocols since they provide strong and provable security and privacy guarantees. Moreover, we integrate differential privacy and cryptographic protocols to achieve strong privacy guarantees efficiently and practically for data processing, management, and analysis.

In data processing, we focus on data anonymization and location data perturbation in supply chains. Data de-identification is essential to comply with privacy laws, such as GDPR, before possible sharing or analysis. We propose an anonymization algorithm by combining differential privacy and k-anonymity to achieve stronger privacy guarantees or better data utility than using them alone. Meanwhile, we consider trajectory hiding under possible attacks and real maps, which propose a more practical solution to share trajectory data under privacy protection.

In data management, we address secure data sharing with cryptographic protocols. Data sharing is vital in data management to advance collaboration and knowledge. However, possible data breaches and malicious inputs can lead to potential financial loss and identity theft. In this thesis, we propose a framework for sharing logistic data in a privacy-preserving way using blockchain and cryptographic protocols. Differential privacy is applied to anonymize data, while cryptographic protocols enhance privacy during data transmission.

In data analysis, we pay attention to privacy-preserving machine learning. Machine learning models are usually trained on large datasets which may contain sensitive personal information. It is important to consider privacy protection during the training and utilization of models. We use differential privacy and secure multi-party computation techniques to design a framework for collaborative learning among multiple parties against inference attacks. Also, we utilize zero-knowledge proof to validate model integrity without leaking the model.

SAMENVATTING

In de moderne maatschappij zijn data steeds belangrijker voor het dagelijks leven van mensen en commerciële activiteiten om organisatie, beheer en analyse ten goede te komen. Correct datagebruik omvat verschillende fasen en de datalevenscyclus wordt geïntroduceerd om de procedures te beschrijven als generatie, verzameling, verwerking, opslag, beheer, analyse, visualisatie en interpretatie. Gedurende de datalevenscyclus is dataprivacy zeer belangrijk, aangezien mogelijke inbreuken kunnen leiden tot misbruik van persoonlijke informatie en financieel verlies. In dit proefschrift bevorderen we de bescherming van dataprivacy in dataverwerking, databeheer en data-analyse gecorreleerd met het *Spark Living Lab*-project op het gebied van toeleveringsketens en machinaal leren. Om de privacybescherming te verbeteren, stellen we oplossingen voor op basis van differentiële privacy- en cryptografische protocollen, aangezien deze sterke en aantoonbare beveiligings- en privacygaranties bieden. Bovendien integreren we differentiële privacy- en cryptografische protocollen om op een efficiënte en praktische wijze sterke privacygaranties te bereiken voor dataverwerking, -beheer en -analyse.

Bij dataverwerking richten we ons op data-anonimisering en locatiegegevensverstoring in toeleveringsketens. Data-anonimisering is essentieel om te voldoen aan privacywetten, zoals de AVG, vóór mogelijke data uitwisseling of analyse. We stellen een anonimiseringsalgoritme voor door differentiële privacy en k-anonimiteit te combineren om sterkere privacygaranties of beter data-nut te bereiken dan wanneer ze alleen worden gebruikt. Ondertussen overwegen we trajectverberging voor mogelijke aanvallen en echte kaarten, wat een praktischer oplossing oplevert om trajectgegevens te delen met privacybescherming.

In databeheer richten we ons op veilige gegevensuitwisseling met cryptografische protocollen. Gegevensuitwisseling is van vitaal belang in databeheer om samenwerking en kennis te bevorderen. Mogelijke datalekken en kwaadaardige invoer kunnen echter leiden tot potentieel financieel verlies en identiteitsdiefstal. In dit proefschrift stellen we een raamwerk voor om logistieke gegevens op een privacybeschermende manier te delen met behulp van blockchain en cryptografische protocollen. Differentiële privacy wordt toegepast om gegevens te anonimiseren, terwijl cryptografische protocollen de privacy tijdens gegevensoverdracht verbeteren.

In data-analyse besteden we aandacht aan privacybeschermende machine learning. Machine learning-modellen worden meestal getraind op grote datasets die gevoelige persoonlijke informatie kunnen bevatten. Het is belangrijk om privacybescherming in acht te nemen tijdens de training en het gebruik van modellen. We gebruiken differentiële privacy en veilige multi-party computation-technieken om een raamwerk te ontwerpen voor samenwerkend leren tussen meerdere partijen tegen inferentieaanvallen. Bovendien maken we gebruik van een zero-knowledge bewijs om de integriteit van het model te valideren zonder het model prijs te geven.

1

INTRODUCTION

In the 21st century, data is playing an essential role in the world, influencing different aspects of our lives, including personalized experience, healthcare improvement and scientific research. According to statistics from *IDC* and *Statista*, the global data volume reached 64.2 zettabytes in 2020 and is predicted to soar to over 180 zettabytes in 2025 [1]. The tremendous growth highlights the increasing influence of data on both individuals and businesses. For individuals, data can improve personal organization, financial management, travel navigation, fitness tracking, social and shopping behaviour analysis. For companies, data-driven insights can contribute to better supply chain management, risk management, market analysis and research, financial analysis and planning. As a result, data has become more indispensable in our lives than ever, and its proper utilization is of greater importance.

Proper data utilization involves different phases. Jeannette Wing, a renowned researcher in cybersecurity and privacy, introduced the data life cycle to illustrate the whole procedure as shown in Figure 1.1, including generation, collection, processing, storage, management, analysis, visualization, and interpretation [2]. In the beginning, data can be generated from people and sensors. People generate data from their daily behaviour, such as surfing websites, while sensors can generate data while monitoring, such as the temperature, humidity and wind of the day. After generation is data collection. Since data streams usually come rapidly, it is impossible to collect only the proportion of data which is valuable for analysis or different tasks. Instead, all data are collected and processed afterwards. Data processing involves possible data cleaning, wrangling, and privacy-preserving approaches such as anonymization. Then, data is structurally stored in hard disks or solid-state drives (SSDs) to optimize possible future uses. However, data comes in different types and velocities. From data management, we need to utilize various kinds of data and share them with other parties if needed. After that, data can be used for analysis in machine learning or data mining, which can help people make decisions. Finally, data visualization and interpretation enable people to better understand the result of data analysis with pictures and explanations.

Along the data life cycle, data privacy should be carefully considered at each phase

1

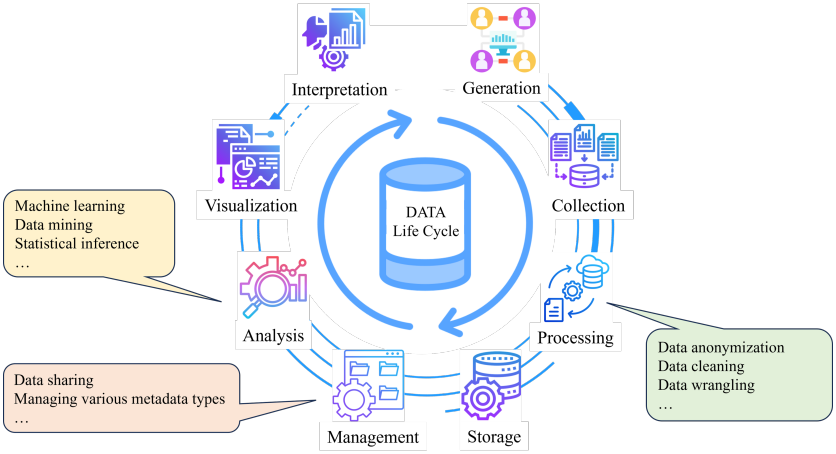


Figure 1.1: Data Life Cycle [2].

since sensitive personal information can be included [2]. Meanwhile, the leakage of commercial data can result in data ownership issues and associated potential commercial value loss. Different countries enforce laws and regulations to enhance data privacy, such as the General Data Protection Regulation (GDPR) in the EU, the Personal Information Protection Law (PIPL) in China, and the California Consumer Privacy Act (CCPA) in the US. Though regulations force companies and organizations to take action to avoid data leakage, in recent years, severe data privacy breaches still occurred. For example, in 2023, 815 million COVID testing records were stolen from the Indian Council of Medical Research [3]. Also, the Facebook breach in 2021 exposed 533 million user records, including names, phone numbers, locations, and email addresses [3]. Such breaches can cause the abuse of personal information or financial loss, which emphasizes the importance and urgency of advancing data privacy protection [4].

There is an increasing demand for data utilization and data privacy protection in different domains. For example, in supply chain management, which ensures the transportation of goods and greatly impacts the whole economy and business, supply chain data is essential for improving the efficiency and performance of supply chain operations, but the data is sensitive and can imply commercial secrets such as the type of goods and the transportation route. Based on that, in 2020, the *Spark! Living Lab* project was granted by the Dutch Research Council. In collaboration among different institutes and logistic companies from both the industry and academia [5], the project aims at supply chain 4.0 and data-driven logistics while utilizing sensitive logistic information in a secure and privacy-preserving way to offer reliable and accurate operational services.

For privacy and security concerns in the project, the main goal of the thesis is to achieve confidentiality, integrity, and availability (CIA). Confidentiality ensures that sensitive supply chain data is only available for authorized users and not accessible to the adversary. Integrity maintains data accuracy and reliability, preventing potential adversarial tampering during data sharing. Availability guarantees the system is reliable and functioning when authorized users access the resources. Such properties are further

considered according to different use cases from the *Spark! Living Lab* project, which ranges from certificate management to goods condition surveillance. In the thesis, the following three use cases are further considered with more specific adversary models.

Use Case 1: Data anonymization for bin-packing. This use case is from one of our partners, the Port of Rotterdam. In logistics, the bin-packing problem is how to load packages into a minimum number of containers, which requires detailed package information for optimization, such as the weight and volume of packages. However, the package data can imply the type or recipients of goods, resulting in the theft of targeted products from the ports [6, 7]. In the adversary model, we assume the adversary is the logistic operator with access to the package data but can misuse the data to identify the target package for theft. As a result, package data anonymization is essential to lower the risk of potential thieves and achieve privacy-preserving bin-packing.

Use Case 2: Privacy-preserving location data sharing. This use case is from the logistics companies in the *Spark! Living Lab* project. In logistics, location data is needed for delivery tracking and border checks. A real-time tracking system can improve the estimation of delivery time and improve customer satisfaction [8]. Also, during transportation, location data is essential for border checks in the *electronic Contract for the International Carriage of Goods by Road* (eCMR) [9]. However, location data can expose the location of truck drivers, which violates GDPR [10], and provides an open gate for package thieves [7]. In our adversary model, we assume the potential adversary can be (1) the internal user who has access to the shared location data and aims to locate the truck for theft and malicious tracking or (2) the external user who wants to steal the location data of the truck without access. The adversaries hold background knowledge of the truck, such as the city's road map. As a result, it is essential to apply location data perturbation and privacy-preserving data sharing. Location data perturbation aims to publish an approximate location of the truck, which is useful for tracking, but the accurate location point is hidden. Privacy-preserving data sharing guarantees that only the recipients can receive the approximate location information but no one else.

Use Case 3: Privacy-preserving machine learning. During the progress of the project, it became apparent that machine learning-based solutions, such as fraud detection and demand forecasting, can benefit productivity and decision-making for supply chains. Such machine learning models are trained on sensitive supply chain data, and it is crucial to train and use a machine learning model in a privacy-preserving manner. In the adversary model, we assume the adversary can be (1) the end users who aim to infer further information about the training data of a machine learning model or (2) the model owners who try to break the integrity of machine learning models by claiming an unachievable high accuracy or a model type which is not actually used. As a result, designing privacy-preserving machine learning solutions can have significant impacts on supply chain management. More specifically, we consider the setting of a small group of small and medium-sized enterprises (SMEs) jointly training a model (to save cost) while protecting against possible data leakage and privacy attacks. When the model is trained, they can sell or provide their model services to others. Then, it is important to prove the ownership, correctness and performance of the private model without leaking the

structure of the model or the training data.

As mentioned above, all use cases rely on sensitive supply chain data. On the one hand, we want to improve supply chain operations based on the available data. On the other hand, we need to avoid sensitive data being misused or stolen. Considering the data life cycle and our use cases, it is vital to develop privacy-preserving solutions for data processing, management, and analysis, which determine how data looks, how it is managed among parties, and how it is used for analysis. Regarding our use cases, we further investigate the procedures in data anonymization for bin-packing, privacy-preserving location data sharing, and privacy-preserving machine learning.

To deal with the privacy issues in our use cases, we rely on cryptographic tools to achieve secure communication among multiple parties against potential adversaries. Cryptographic tools provide strong and provable security and privacy guarantees, but protocols based on such tools suffer from runtime efficiency issues. Differential privacy is another powerful tool for privacy protection. It also provides privacy guarantees with high efficiency, but it is challenging to find a good utility-privacy trade-off. In this thesis, we utilize and integrate cryptographic protocols and differential privacy for more advanced solutions for different use cases.

In the rest of the chapter, we first provide a brief description of the privacy-enhancing techniques we utilized in this thesis in Section 1.1. Then, we discuss the privacy challenges in Section 1.2 in more detail with respect to our use cases. After that, we proceed with our problem statement in Section 1.3 and finalize the chapter with our contributions in Section 1.4.

1.1. PRIVACY ENHANCING TECHNIQUES

Differential privacy, k-anonymity, and cryptographic protocols provide hands-on approaches to privacy preservation. This section further explains their concepts and how they can benefit data privacy. In brief, differential privacy applies small noise to a query or dataset to protect data privacy [11, 12]. k-anonymity achieves data anonymization using data generalization and suppression [13]. In contrast, modern cryptographic protocols are based on mathematical principles to achieve communication in the presence of adversaries [14]. An adversary is malicious and aims to hinder the users of the cryptosystem from attaining their objectives. The following provides details and the historical background of differential privacy and cryptographic protocols.

DIFFERENTIAL PRIVACY

Differential privacy [11, 12, 15] provides strong privacy guarantees for algorithms over aggregate datasets, which implies that the existence of any record in the dataset does not influence the output probability with factors ϵ and δ. This property prevents adversaries from determining whether a specific sample is included in the dataset or not.

Definition 1.1 ((ϵ,δ)-differential privacy)**.** *A randomized algorithm $\mathcal{M}$ satisfies (ϵ,δ)-differential privacy if for all neighboring datasets $\mathcal{D},\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}$ differing in one element, and any $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$,*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta, \tag{1.1}$$

where e is the base of the natural logarithm function, $\mathbb{N}$ is the set of non-negative integers and $\mathcal{X}$ is the universe for all datasets. If δ is 0, $\mathcal{M}$ satisfies ϵ-differential privacy.

Among plenty of differentially private mechanisms, the Laplace mechanism [15] and the exponential mechanism [16] are widely applied, and they achieve differential privacy with different approaches. The Laplace mechanism is usually used to deal with numeric data and queries. The Laplace noise is added to the accurate output of the query f so that the output of $\mathcal{M}$ satisfies ϵ-differential privacy over the query. In contrast, the exponential mechanism aims to select one item from a possible finite output group. For each item, a utility score is determined by a predefined utility function. A probability is then assigned to each possible output according to the score. By doing that, the mechanism can output a precise element using the probabilities, and perturbation is added for the selection procedure.

Other mechanisms, such as the Gaussian mechanism [15] and the geometric mechanism [17], achieve differential privacy similar to the Laplace mechanism, but the Gaussian mechanism applies the Gaussian noise, and the geometric mechanism works with integer values. Besides, similar to the exponential mechanism, the permute-and-flip [18] randomly chooses a value from a set of options, weighed by a utility score and the privacy parameter ϵ. For each item, a biased coin is flipped, and this item is returned if the coin comes with a head. If the item is not returned, the probability of outputting the next item is updated. The authors of [18] show that the permute-and-flip mechanism never performs worse than the exponential mechanism.

Differential privacy can provide strong privacy guarantees in practice. In data anonymization, differentially private mechanisms apply noise to the raw dataset or during the sampling so that an adversary can not identify individuals' information from the output dataset [19–21]. Also, differential privacy is widely applied in machine learning models against inference attacks, which aim to identify whether a specific record is in the training set or not. The noise is applied to the gradient or different processes during the training [22–24]. What's more, differential privacy is applied to other domains, such as location privacy [25], recommender systems [26], and data mining [27].

k-ANONYMITY

k-anonymity is a widely used approach for data anonymization [13]. For each record in the dataset, there are at least other $k-1$ same records so that the record is indistinguishable. While achieving optimal k-anonymity is NP-hard, there are plenty of effective solutions. Mondrian [28] applies *kd-tree* to split the dataset and reconstruct it with equivalence classes whose size is at least k. Mondrian works well with numerical data but not on categorical ones. Also, Emam et al. [29] proposed OLA, which achieves k-anonymity using a pre-defined generalization hierarchy with generalization rules for each attribute. The approach works well with numerical and categorical data but suffers from high computational complexity. Other k-anonymous approaches, such as k-Optimize [30], Flash [31], Datafly [32], achieve k-anonymity in different ways, but either the run time is long for large datasets or too much utility is lost during the process. Besides the efficiency problem, another problem for k-anonymity is the lack of mathematical and unambiguous proof of which attributes are identifiable, which needs further investigation and studies.

1

CRYPTOGRAPHIC PROTOCOLS

Cryptography is about achieving secure communication with the presence of adversaries [14]. Modern cryptography algorithms are widely used to hide secrets and provide secure communication among computer networks, but more advanced and complicated protocols are still needed to achieve different goals. In this thesis, when designing a privacy-preserving collaborative learning framework, we utilize **homomorphic encryption** to compute a function over multiple parties while keeping their inputs private. Also, we apply **zero-knowledge proof (ZKP)** to prove the correctness and performance of a machine learning model in a privacy-preserving manner. Now we briefly introduce these two concepts.

HOMOMORPHIC ENCRYPTION

Homomorphic encryption (HE) [33] is a crucial approach for secure multi-party computation (MPC), which was first introduced by Yao in 1982 to consider the Millionaires' Problem where two millionaires want to know who is richer without exposing any information about their wealth [34]. HE is a public key encryption scheme where mathematical operations, such as addition and multiplication, can be performed on the ciphertext with the result correct and encrypted. As a result, HE allows a third party to perform calculations without accessing the raw data. According to the type and number of operations, there are three types: *partially homomorphic encryption* that supports either addition or multiplication; *somewhat homomorphic encryption* that supports a limited number of additions and multiplications; and *fully homomorphic encryption* that supports an unlimited number of addition and multiplication operations [35]. For example, for partially homomorphic encryption, in an additive homomorphic encryption scheme, such as Paillier [36], we have

$$\mathbf{D}_{sk}(\mathbf{E}_{pk}(m_1) \cdot \mathbf{E}_{pk}(m_2)) = m_1 + m_2, \tag{1.2}$$

where $\mathbf{E}_{pk}(m)$ denotes the encryption of m using the public key pk, and $\mathbf{D}_{sk}(c)$ denotes the decryption of c using the secret key sk. Similarly, in a multiplicative homomorphic encryption scheme, such as ElGamal [37], we have

$$\mathbf{D}_{sk}(\mathbf{E}_{pk}(m_1) \cdot \mathbf{E}_{pk}(m_2)) = m_1 \cdot m_2. \tag{1.3}$$

HE schemes, including RSA [38], ElGamal [37], Paillier [36], Gentry's FHE scheme [39], are applied in different domains such as e-voting [40, 41], data analysis [42, 43], and cloud computing [44–46]. Nevertheless, further study is needed to improve the efficiency (to lower the computational overhead) while keeping the scheme secure [35].

ZERO-KNOWLEDGE PROOF

Different from MPC, zero-knowledge proof allows a *prover* to prove to a *verifier* that a statement is true without revealing any information beyond the validity of the statement [47]. To achieve this goal, the protocol demands interactions between the prover and the verifier, but this suffers expensive communication costs. Then, the idea of non-interactive zero-knowledge proof (NIZK) is proposed to avoid interactions [48, 49]. In an NP relationship $\mathcal{R}$, the verifier $\mathcal{V}$ can verify a statement from the prover $\mathcal{P}$ based on the proof π with a witness w for input x with $(x; w) \in \mathcal{R}$. In general, a zero-knowledge proof scheme has the properties of completeness, soundness, and zero-knowledge.

- Completeness. For any true statement, the probability is 1 that the proof from an honest prover can convince an honest verifier, which means that with the public parameters pp and $\pi \leftarrow \mathcal{P}(x, w, \text{pp})$, if $(x; w) \in \mathcal{R}$, we have

$$\Pr[\mathcal{V}(x, \pi, \text{pp}) = 1] = 1. \tag{1.4}$$

- Soundness. For any false statement, the probability is negligible that the proof from a malicious prover can deceive an honest verifier, which means that for any probabilistic polynomial time (PPT) prover $\mathcal{P}^*$ and a PPT extractor $\mathcal{E}$, they can extract a witness w that $\pi^* \leftarrow \mathcal{P}^*(x, \text{pp})$ and $w \leftarrow \mathcal{E}^{\mathcal{P}^*}(\pi^*, x, \text{pp})$. If $(x; w) \notin \mathcal{R}$, we have

$$\Pr[\mathcal{V}(x, \pi^*, \text{pp}) = 1] < \text{negl}(\lambda). \tag{1.5}$$

- Zero-knowledge. The verifier can not learn any information about the statement beyond its validity, which means that for any PPT verifier $\mathcal{V}^*$ and PPT simulator $\mathcal{S}$, if $(x; w) \in \mathcal{R}$, we have

$$\text{View}(\mathcal{V}^*(x, \text{pp})) \approx \mathcal{S}^{\mathcal{V}^*}(x) \tag{1.6}$$

  where View denotes the view of the verifier during the process, $\mathcal{S}^{\mathcal{V}^*}(x)$ is the view from the simulator with input x and $\mathcal{V}^*$, and $\approx$ implies indistinguishability.

Modern zero-knowledge proof schemes, such as Zero-Knowledge Succinct Non-interactive Argument of Knowledge (zk-SNARKS) [50–53], Bulletproofs [54], are widely used for cryptocurrencies like *Monero* and *Zcash*. Meanwhile, zero-knowledge proofs also benefit other domains, including authentication systems [55, 56], supply chain management [57, 58], and machine learning model evaluation [59–63].

1.2. DATA PRIVACY CHALLENGES

After introducing the concepts of differential privacy and cryptographic protocols, this section describes the privacy issues according to our three use cases along the data life cycle: (1) data anonymization for bin-packing, (2) privacy-preserving location data sharing, and (3) privacy-preserving machine learning. The state-of-the-art privacy-preserving solutions and their limitations are discussed. After that, data privacy challenges are elaborated for each stage with the potential for improvement.

DATA ANONYMIZATION FOR BIN-PACKING

As discussed in **Use Case 1**, during bin-packing, the package data is sensitive and can be used to infer the contents and recipients of the package, so data anonymization is needed for privacy-preserving bin-packing. Data anonymization techniques [11, 13] can prevent possible adversaries from identifying whether an individual is in the dataset after a sensitive dataset is published. According to GDPR, it is not allowed to process identifiable personal data that is related to a natural person without consent. For data utilization, properly anonymized data is crucial for bin-packing, while its influences on overloading or oversizing need to be evaluated.

As introduced in Section 1.1, k-anonymity and differential privacy provide promising approaches for data anonymization by suppression and generalization or by applying a

small noise to the data. However, they have limitations. For k-anonymity, an adversary can carry out linkage attacks to identify an individual in the dataset using background knowledge or comparing the overlap among different datasets. Against such problems, differential privacy is efficient and provides provable strong privacy guarantees, but it can greatly lower the utility of the dataset to achieve privacy protection. One possible way is to combine k-anonymity and differential privacy to optimize the trade-off between privacy and utility, as suggested in [19]. However, no specific approach has been proposed, and it is still an open question to design a feasible solution.

PRIVACY-PRESERVING LOCATION DATA SHARING

As discussed in **Use Case 2**, privacy-preserving location data sharing is critical for delivery tracking and border checks. It involves two parts: (1) location data perturbation and (2) privacy-preserving data sharing.

Location data perturbation, or trajectory hiding, enables the share of an approximate location, which is useful for location-based services but does not leak the accurate location point of the truck. Though k-anonymity can deal with dataset anonymization, it is unsuitable for other types of sensitive data requiring de-identification, such as people's geolocation and trajectory data. Instead, differential privacy perturbs location data and trajectories with privacy guarantees by adding noise to the actual. Existing solutions theoretically consider differential privacy for location point hiding [25] and trajectory hiding [64, 65]. Andrés et al. [25] defined geo-indistinguishability to consider the privacy of individuals within a radius r, but only single location points are protected. Fang et al. [64] introduced a threshold of δ-neighbourhood for continuous location points. Xiao et al. [65] also considered the temporal correlation with Markov chains. However, practical performance is not evaluated under possible attacks if the adversary has background knowledge, such as the road map. Under differential privacy, theoretical privacy guarantees are different from practical protection [66]. Further investigations and research remain to show how well differential privacy protects location privacy in practice.

Data sharing is vital in data management for advancing knowledge, expanding data access and collaborating research. It improves transparency and accountability among different parties, which contributes to better efficiency and improvement in research and business. Nevertheless, companies still avoid sharing their data access with others due to privacy concerns [67]. The shared data usually contains sensitive information about individuals or companies. Different companies lack trust in each other, and data sharing exposes sensitive information to possible unauthorized access, data breaches, misuse and loss [68]. Most existing data-sharing solutions for supply chains are based on centralized services, but centralized systems cannot provide entirely trustworthy data computing, traceability or immutable data storage. Meanwhile, it is costly for SMEs to build their centralized solution. In contrast, blockchain is a distributed and immutable ledger with traceability, non-repudiation, and transparency [69]. Such properties are important to protect against counterfeit products and improve transparency. Also, the blockchain-based platform can be shared among SMEs to save costs and enhance cooperation. Data validation and privacy are of great concern in a blockchain-based supply chain [70]. Since location data is commercially sensitive, and enterprises do not want their data known to other participants, a privacy-preserving blockchain-based data shar-

ing platform is desired [71].

As blockchain can build trust among companies since it is decentralized, traceable, immutable and transparent [69], it is already possible to see blockchain as a data-sharing platform among companies in supply chain management [72–75]. For location sharing, Wu *et al.* [72] propose a framework with multiple private ledgers and one public ledger. The private ledger is used for customers of a specific shipment to share custody events. The public ledger is for global tracking, including the geolocation of trucks. The consensus is based on proof of work, and the load is increased due to private and public chains. Privacy is considered in the private ledger, but location perturbation is not addressed. Furthermore, data validation is done by crowd-sourcing from the participants in the private chain to ensure that they share the same information. This can not deal with malicious users who add fake data or events to the chain.

Though blockchain provides possible solutions for data sharing among companies, there are two challenges remaining: the validation of the source of data and possible data breaches on the transparent ledger. It is challenging to achieve data validation to avoid maliciously inserted fake data with human participation, and most existing blockchain-based supply chain solutions do not address the challenge [72–75]. To deal with it, one possible way is to avoid human interaction and enable IoT devices to sign the data when collected, but this is still not well addressed in the existing works. Data breaches can occur during data sharing due to poor access control or the property of transparency. Cryptographic protocols can be applied to allow only authorized users to access the data or transactions in the blockchain. However, existing research has not paid much attention to the privacy of data sharing with blockchain. Privacy preservation is considered in DECOUPLES [74], but the design lacks a general framework and feasibility for different blockchain platforms.

PRIVACY-PRESERVING MACHINE LEARNING

As discussed in **Use Case 3**, we focus on privacy-preserving machine learning (ML) in terms of privacy-preserving small-scale collaborative learning and private machine learning model validation. In ML, adversaries aim to gain sensitive information about the training set or the ML model. Based on the work [76], privacy attacks on ML models can be categorized into four types: (1) membership inference attacks, which aim to determine whether a specific entry was included in the training set, (2) property inference attacks, which aim to infer additional information that only holds for a subset of the training set and was not included as a feature, (3) reconstruction attacks, which aim to reconstruct parts or whole of the training samples, and (4) model extraction attacks, which aim to create a substitute model with very similar functioning to the target model.

In collaborative learning, where a group of parties jointly train a model without sharing data in a parallel or sequential manner, the adversary can deploy inference attacks to exploit the model to infer personal information without directly accessing the underlying data, which can cause serious privacy leakage [77]. Collaborative learning deals with the lack of training data while avoiding sharing data among different parties, but it also suffers from membership and property inference attacks. To perform membership inference attacks, a shadow model and an attack model are trained [78]. The shadow model is trained on data similar to the target model so that predictions by the shadow model

are similar to the target model. The attack model is trained using predictions from the shadow model with ground truth labels, indicating whether the used sample was originally included in the training set [78]. Property inference attacks are performed similarly, except that the data is labelled according to the inferred property [77].

Differential privacy mitigates inference attacks by applying a certain amount of noise at a record or participant level during the training. At the record level, noise is applied to obfuscate individual records. Since the noise is identically distributed for all dataset entries, record-level differential privacy protects against membership inference attacks but not property inference. Instead, participant-level differential privacy ensures that the model behaviour does not change much when all data entries from a specific user are removed [79]. An adversary is unable to determine which data point belongs to which user. Thus, participant-level differential privacy prevents property inference attacks. Unfortunately, to maintain both good model utility and privacy protection, thousands of participants are required [77]. As noted in [80], more research is needed to prevent property inference attacks in collaborative settings with low numbers of participants. A potential approach is to apply efficient cryptographic protocols or combine them with differential privacy to protect against inference attacks while maintaining data utility.

Apart from data privacy for the training data, the ownership and integrity of a machine learning model are critical. The model can have high values and include sensitive private information on the training data. While machine learning applications are widely used nowadays, integrity is a concern that a model is stated to have high accuracy while it is not reproducible and can not be validated [59]. It is necessary to validate a model by testing it on public datasets and checking that the prediction is correctly computed by the model. However, the model owner is reluctant to release the model publicly since this leaks the model itself. For models with high values or high ownership requirements, publishing the model is not a choice because an adversary can simply take the model for personal use. To address the problem of integrity while protecting the privacy of models, one solution is zero-knowledge proof. The prover sends proof that the model correctly generates the prediction with a given dataset, while the verifier can evaluate the prediction and check the correctness of computing with the proof [59]. There are existing solutions for zero-knowledge machine learning models, such as decision trees [59] and neural networks [60–63]. However, as the existing work assumes that the model is correctly structured, a more robust security model is needed. Also, a scheme with lower storage or computation costs is needed for large-scale datasets in practical and real-world uses. One way is to formulate the machine learning model into a format which is more efficient for zero-knowledge proof to accelerate and optimize the zero-knowledge proof process.

1.3. PROBLEM STATEMENT

The previous sections include the background and importance of data privacy in the domains of supply chain management and machine learning according to the use cases in the *Spark! Living Lab* project. Considering the data privacy challenges discussed in Section 1.2, existing data anonymization techniques for bin-packing have high computation costs, and their influence on overloading and oversizing is not well evaluated. For location data sharing, the existing adversary model is not practical, and there is a lack

of a general framework for location data sharing among trucks, recipients and logistic companies. For machine learning, privacy preservation in collaborative learning among small-scale parties is not well addressed in existing works. Also, it is still time-consuming to validate the correctness and performance of a decision tree model. The research goal of the thesis is to mitigate the stated problems in the use cases in supply chain management and machine learning to enhance data privacy protection during bin-packing, location data sharing, and machine learning.

Considering the data privacy protection methods introduced in Section 1.1, cryptographic protocols provide provable security and privacy guarantees, and differential privacy is efficient and measurable for privacy protection. Both methods are robust ways of protecting data privacy while keeping the utility. This thesis applies differential privacy and cryptographic protocols to advance privacy protection techniques. Meanwhile, we further introduce our research questions according to the use cases.

DATA ANONYMIZATION FOR BIN-PACKING

As described in Section 1.2, data anonymization for bin-packing is essential to comply with GDPR for data processing. However, existing solutions mainly rely on k-anonymity, which can take hours for anonymization. Meanwhile, the utility of the output k-anonymous dataset can be low if the optimal is not found. Moreover, k-anonymity does not give theoretical proof on how to choose identifiable attributes, and it suffers from possible linkage attacks with background knowledge. For bin-packing, it is also not well studied how often packing violation (overloading or oversizing) happens when anonymization techniques are applied. The limitations of existing solutions imply the need to increase the efficiency in terms of anonymization time and space usage for privacy-preserving bin-packing, which brings the first research question:

Q1: *How to increase efficiency in privacy-preserving bin-packing?*

PRIVACY-PRESERVING LOCATION DATA SHARING

As discussed in Section 1.2, privacy-preserving location data sharing involves location data perturbation and data sharing. Geolocation data is highly sensitive since it implies the physical location of humans. Existing solutions apply differential privacy as the basic approach to sanitize trajectory data of people or vehicles, but a more practical adversary model is not considered where the adversary can carry out possible attacks and hold background knowledge about the trajectory, such as the road map of the city. This brings the second research question:

Q2: *How to provide location privacy for tracking services for logistics in practice?*

For data sharing, as discussed in Section 1.2, blockchain provides a decentralized, traceable, immutable and transparent data-sharing platform for multiple parties to share their data. However, the privacy concerns in such decentralized platforms are usually not well addressed, such as how the data is only available to the responding users without information leakage. For data validation, the constrained IoT device is cheap and can avoid human interaction, but its capability and battery life remain unknown if location privacy algorithms and cryptographic protocols are applied. There is a lack of a

general framework on how data can be shared under privacy preservation and data validation using a decentralized platform with constrained IoT devices among the trucks, the logistic companies and the users. This brings our third research question:

Q3: *How to design a privacy-preserving and decentralized platform for location sharing with constrained IoT devices?*

PRIVACY-PRESERVING MACHINE LEARNING

Section 1.2 argues that data analysis is the core of data science, and machine learning is the core of data analysis. Though machine learning brings lots of benefits to our lives, the model is trained on possibly highly sensitive data, which emphasizes the importance of enhancing data privacy in machine learning. There are a large number of papers protecting against possible attacks using differential privacy and cryptographic protocols. Further research remains to be done to achieve higher utility while preserving data privacy during the training process. As discussed in Section 1.2, though participant-level differential privacy can protect against property inference attacks in large-scale collaborative learning, it can bring too much noise and result in low utility in a setting among a small number of participants. This brings the fourth research question:

Q4: *How to design a privacy-preserving small-scale collaborative learning system against inference attacks?*

Another critical privacy issue for machine learning is the privacy or ownership of the model itself. A machine learning model may include sensitive information about the owner, or the model may have a high value. It is important to validate the integrity of the model while keeping it secret from adversaries. Zero-knowledge proof is a strong candidate to prove that the model correctly generates the prediction while not leaking any information about the model. There are limited existing solutions for zero-knowledge machine learning models. Most of them are for neural networks, and only a few are for decision trees. Further studies are needed to improve the storage and computation cost while considering stronger security assumptions for a more powerful adversary for zero-knowledge decision trees. This brings our fifth research question:

Q5: *How to validate the correctness and performance of a private decision tree in a privacy-preserving manner?*

1.4. CONTRIBUTION OF THE THESIS

The thesis consists of seven chapters and an appendix. From Chapter 2 to Chapter 6, each chapter is composed of one research paper. Since different papers are published or submitted to different venues, all chapters can be read separately. The notations in different chapters may conflict, but a notation table is included in each chapter. The technical details of the papers are kept with possible minor revisions compared to the published version.

CHAPTER 2
DATA ANONYMIZATION

This chapter answers to the research question **Q1**. In this chapter, we address the data anonymization issue in bin-packing, where data such as the weight of the packages is needed when assigning items to trucks. However, the information is sensitive and can be used to identify the contents of the package. To protect privacy during bin-packing, we propose two different privacy-preserving data publishing methods. Both approaches use differential privacy (DP) to hide the existence of any specific package to prevent it from being identified by adversaries. The first approach combines differential privacy with k-anonymity, and the other one applies clustering before differential privacy. We show that the proposed approaches have better privacy guarantees, better efficiency, and better performance than the existing works that use either differential privacy or k-anonymity. This chapter is a copy of the paper titled "Privacy-Preserving Bin-Packing with Differential Privacy" by Li, T., Erkin, Z., and Lagendijk, R. L. in *IEEE Open Journal of Signal Processing* vol. 3, pp. 94-106, 2022.

CHAPTER 3
LOCATION DATA PERTURBATION

This chapter answers to the research question **Q2**. In this chapter, we propose a privacy-preserving real-time location sharing system, including a differential privacy-based location publishing method and location sharing protocols for both centralized and decentralized platforms. Different from existing location perturbation solutions, which only consider privacy in theory, our location publishing method is based on a real map and different privacy levels for recipients. With analyses and proofs, the proposed location publishing method provides better privacy protection than existing works under real maps against possible attacks. We provide a detailed analysis of the choice of the privacy parameter and its impact on the suggested noisy location outputs. This chapter is a copy of the paper titled "Trajectory Hiding and Sharing for Supply Chains with Differential Privacy" by Li, T., Xu, L., Erkin, Z., and Lagendijk, R. L. in *28th European Symposium on Research in Computer Security (ESORICS 2023)*, pp. 297-317, 2023.

CHAPTER 4
LOCATION DATA SHARING WITH PRIVACY PRESERVATION

This chapter answers to the research question **Q3**. In this chapter, we propose a blockchain-based supply chain solution with location privacy preserved on trucks for multiple participants. We use cryptographic tools to construct protocols for our platform framework, ensuring secure and privacy-preserving data sharing. Meanwhile, we deploy our efficient differentially private location privacy algorithm for constrained IoT devices embedded in the delivery truck to avoid human interaction and possible physical intrusion. Data validation and transparency are achieved within the supply chain and logistics. Our evaluation demonstrates the practicality of deploying our location privacy algorithm on constrained IoT devices in real-world scenarios. This chapter is a copy of the paper titled "PrivTrack: Privacy-Preserving Trajectory Tracking for Supply Chains" by Li, T., Kromes, R., Erkin, Z., and Lagendijk, R. L., which is under review from *IEEE Transactions on De-*

pendable and Secure Computing.

CHAPTER 5
DATA PRIVACY ENHANCEMENT FOR COLLABORATIVE LEARNING

This chapter answers to the research question **Q4**. In this chapter, we propose a novel protocol leveraging secure multi-party computation and differential privacy to prevent inference attacks in sequential collaborative learning. Participants jointly determine a training order, and they only receive information on whom to send their data, so they are unaware of whose data they are receiving. We prevent inference attacks using a secure joint permutation selection protocol with an overhead of only a few seconds. Meanwhile, we apply differential privacy to hide the training order by applying a random time delay. This chapter is a copy of the paper titled "Robust Small-Scale Collaborative Learning Against Inference Attacks" by Li, T., van Tetering, D., and Erkin, Z., which is under review from *IEEE Transactions on Information Forensics and Security*.

CHAPTER 6
ZERO-KNOWLEDGE DECISION TREE

This chapter answers to the research question **Q5**. In this chapter, we give a novel application of zero-knowledge matrix lookup arguments to the domain of zero-knowledge decision trees. The model owner can prove the performance and correctness of a decision tree by releasing a commitment and proving zero-knowledge statistics over the committed data structure. Our scheme based on lookup arguments has succinct verification. The prover's time complexity is asymptotically better than the state of the art and is secure in a strong security model where the commitment to the decision tree can be malicious. This chapter is a copy of the paper titled "Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees" by Campanelli, M., Faonio, A.*, Fiore, D., Li, T.*, and Lipmaa, H. (in alphabetical order), in *Public-Key Cryptography (PKC 2024)*, pp. 337-369, 2024.

CHAPTER 7
DISCUSSION

This chapter discusses the answer to the main research question. This chapter summarizes the conclusion of the thesis and the future work according to the findings in Chapter 2 to Chapter 6. Based on that, this chapter addresses how much and how well the research question is answered in the thesis.

In summary, here we list the publications that are included in the main chapters of the thesis.

1. **Li, T.**, Erkin, Z., and Lagendijk, R. L. "Privacy-Preserving Bin-Packing with Differential Privacy", in *IEEE Open Journal of Signal Processing* vol. 3, pp. 94-106, 2022.

2. **Li, T.**, Xu, L., Erkin, Z., and Lagendijk, R. L. "Trajectory Hiding and Sharing for Supply Chains with Differential Privacy", in *28th European Symposium on Research in Computer Security (ESORICS 2023)*, pp. 297-317, 2023.

3. **Li, T.**, Kromes, R., Erkin, Z., and Lagendijk, R. L. “PrivTrack: Privacy-Preserving Trajectory Tracking for Supply Chains", under review from *IEEE Transactions on Dependable and Secure Computing*.

4. **Li, T.**, van Tetering, D., and Erkin, Z. “Robust Small-Scale Collaborative Learning against Inference Attacks Using Multi-Party Selection and Differential Privacy", under review from *IEEE Transactions on Information Forensics and Security*.

5. Campanelli, M., Faonio, A.*, Fiore, D., **Li, T.***, and Lipmaa, H. (in alphabetical order), “Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees”, in *Public-Key Cryptography – PKC 2024*, pp. 337-369, 2024.

Also, the following publications were published during the PhD study but are not included in the thesis since they are less relevant to the topics developed in this thesis.

6. **Li, T.**, Vos, J., and Erkin, Z. “Decentralized Private Freight Declaration & Tracking with Data Validation”, in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom 2022 Workshops)*, pp. 267-272, 2022.

7. Kromes R., **Li, T.**, Bouillon, M., Güler, T. E., Hulst, V., and Erkin, Z. “Fear of Missing Out: Constrained Trial of Blockchain in Supply Chain”, in *Sensors* 24(3), pp. 986-1010, 2024.

8. Xu, L., **Li, T.**, and Erkin, Z. “Verifiable Credentials with a Privacy-Preserving Tamper-Evident Revocation Mechanism", in *Fifth International Conference on Blockchain Computing and Applications (BCCA 2023)*, pp. 266-273, 2023.

9. Kester, D., **Li, T.**, and Erkin, Z. “PRIDE: A Privacy-Preserving Decentralised Key Management System”, in *2022 IEEE International Workshop on Information Forensics and Security (WIFS 2022)*, pp. 1-6, 2022.

10. Vos, D., Vos, J., **Li, T.**, Erkin, Z., and Verwer, S. “Differentially-Private Decision Trees with Probabilistic Robustness to Data Poisoning", *arXiv preprint*, 2023.

1

REFERENCES

[1] IDC and Statista. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 (in zettabytes)*. https://www.statista.com/statistics/871513/worldwide-data-created/. Accessed: 2024-03-05. 2021.

[2] J. M. Wing. "The Data Life Cycle". In: *Harvard Data Science Review* 1.1 (July 2019).

[3] M. Komnenic. *109 Biggest Data Breaches, Hacks, and Exposures [2024 Update]*. https://termly.io/resources/articles/biggest-data-breaches/. Accessed: 2024-03-05. 2023.

[4] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki, D. Margolis, V. Paxson, and E. Bursztein. "Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 1421–1434.

[5] Consortium. *Spark! Living Lab Project*. https://sparklivinglab.nl. Supported by the Dutch Research Council (NWO) under Project 439.18.453B. 2020-2024.

[6] A. Van den Engel and E. Prummel. "Organised theft of commercial vehicles and their loads in the European Union". In: *European Parliament. Directorate General Internal Policies of the Union. Policy Department Structural and Cohesion Policies. Transport and Tourism, Brussels* (2007).

[7] M. Essig, M. Hülsmann, E.-M. Kern, and S. Klein-Schmeink. *Supply chain safety management*. Springer, 2013.

[8] M. Grmiling. *How Real Time Tracking Can Improve Logistics*. https://www.hublock.io/how-real-time-tracking-can-improve-logistics/. Accessed: 2021-11-17. 2021.

[9] J. C. Ferrer. "The CMR Convention - A Pillar of International Carriage of Goods by Road (Abstract)". In: *Uniform Law Review* 11.3 (2006), pp. 521–521. ISSN: 1124-3694. eprint: https://academic.oup.com/ulr/article-pdf/11/3/521/4717296/11-3-521.pdf.

[10] E. Murati and M. Henkoja. "Location data privacy on MaaS under GDPR". In: *Eur. J. Privacy L. & Tech.* (2019), p. 115.

[11] C. Dwork. "Differential Privacy". In: *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 1–12.

[12] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 265–284.

[13] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Technical Report, SRI International. 1998.

[14] R. L. Rivest. "Cryptography". In: *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*. Elsevier and MIT Press, 1990, pp. 717–755.

[15] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.

[16] F. McSherry and K. Talwar. "Mechanism Design via Differential Privacy". In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*. IEEE Computer Society, 2007, pp. 94–103.

[17] A. Ghosh, T. Roughgarden, and M. Sundararajan. "Universally utility-maximizing privacy mechanisms". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. ACM, 2009, pp. 351–360.

[18] R. McKenna and D. Sheldon. "Permute-and-Flip: A new mechanism for differentially private selection". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020.

[19] N. Li, W. H. Qardaji, and D. Su. "On sampling, anonymization, and differential privacy or, *k*-anonymization meets differential privacy". In: *7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012*. ACM, 2012, pp. 32–33.

[20] J. Domingo-Ferrer and J. Soria-Comas. "From t-closeness to differential privacy and vice versa in data anonymization". In: *Knowl. Based Syst.* 74 (2015), pp. 151–158.

[21] N. Holohan, S. Antonatos, S. Braghin, and P. M. Aonghusa. "(k, ϵ)-Anonymity: k-Anonymity with ϵ-Differential Privacy". In: *CoRR* abs/1710.01615 (2017). arXiv: 1710.01615.

[22] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. ACM, 2016, pp. 308–318.

[23] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor. "Federated Learning With Differential Privacy: Algorithms and Performance Analysis". In: *IEEE Trans. Inf. Forensics Secur.* 15 (2020), pp. 3454–3469.

[24] A. Triastcyn and B. Faltings. "Bayesian Differential Privacy for Machine Learning". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9583–9592.

[25] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. “Geo-indistinguishability: differential privacy for location-based systems”. In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. ACM, 2013, pp. 901–914.

[26] F. McSherry and I. Mironov. “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, Paris, France, June 28 - July 1, 2009*. ACM, 2009, pp. 627–636.

[27] A. Friedman and A. Schuster. “Data mining with differential privacy”. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*. ACM, 2010, pp. 493–502.

[28] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. “Mondrian Multidimensional K-Anonymity”. In: *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*. IEEE Computer Society, 2006, p. 25.

[29] K. E. Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. “Research Paper: A Globally Optimal k-Anonymity Method for the De-Identification of Health Data”. In: *J. Am. Medical Informatics Assoc.* 16.5 (2009), pp. 670–682.

[30] R. J. B. Jr. and R. Agrawal. “Data Privacy through Optimal k-Anonymization”. In: *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*. IEEE Computer Society, 2005, pp. 217–228.

[31] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. “Flash: Efficient, Stable and Optimal K-Anonymity”. In: *2012 International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, and 2012 International Confernece on Social Computing, SocialCom 2012, Amsterdam, Netherlands, September 3-5, 2012*. IEEE Computer Society, 2012, pp. 708–717.

[32] L. Sweeney. “Datafly: A System for Providing Anonymity in Medical Data”. In: *Database Securty XI: Status and Prospects, IFIP TC11 WG11.3 Eleventh International Conference on Database Security, 10-13 August 1997, Lake Tahoe, California, USA*. Vol. 113. IFIP Conference Proceedings. Chapman & Hall, 1997, pp. 356–381.

[33] R. L. Rivest, L. Adleman, M. L. Dertouzos, et al. “On data banks and privacy homomorphisms”. In: *Foundations of secure computation* 4.11 (1978), pp. 169–180.

[34] A. C. Yao. “Protocols for Secure Computations (Extended Abstract)”. In: *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. IEEE Computer Society, 1982, pp. 160–164.

[35] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. “A Survey on Homomorphic Encryption Schemes: Theory and Implementation”. In: *ACM Comput. Surv.* 51.4 (2018), 79:1–79:35.

[36] P. Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 223–238.

[37] T. E. Gamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *Advances in Cryptology, Proceedings of CRYPTO ’84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*. Vol. 196. Lecture Notes in Computer Science. Springer, 1984, pp. 10–18.

[38] R. L. Rivest, A. Shamir, and L. M. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (1978), pp. 120–126.

[39] C. Gentry. “A fully homomorphic encryption scheme”. PhD thesis. Stanford University, USA, 2009.

[40] K. Peng, R. Aditya, C. Boyd, E. Dawson, and B. Lee. “Multiplicative Homomorphic E-Voting”. In: *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*. Vol. 3348. Lecture Notes in Computer Science. Springer, 2004, pp. 61–72.

[41] Y. Wu and S. Kasahara. “Smart Contract-Based E-Voting System Using Homomorphic Encryption and Zero-Knowledge Proof”. In: *Applied Cryptography and Network Security Workshops - ACNS 2023 Satellite Workshops, ADSC, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, SiMLA, Kyoto, Japan, June 19-22, 2023, Proceedings*. Vol. 13907. Lecture Notes in Computer Science. Springer, 2023, pp. 67–83.

[42] W. Lu, S. Kawasaki, and J. Sakuma. “Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data”. In: *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*. The Internet Society, 2017.

[43] C. Dhasarathan, M. K. Hasan, S. Islam, S. Abdullah, U. A. Mokhtar, A. R. Javed, and S. Goundar. “COVID-19 health data analysis and personal data preserving: A homomorphic privacy enforcement approach”. In: *Comput. Commun.* 199 (2023), pp. 87–97.

[44] N. Wang, W. Zhou, J. Wang, Y. Guo, J. Fu, and J. Liu. “Secure and Efficient Similarity Retrieval in Cloud Computing Based on Homomorphic Encryption”. In: *IEEE Trans. Inf. Forensics Secur.* 19 (2024), pp. 2454–2469.

[45] A. Hosseingholizadeh, F. Rahmati, M. Ali, H. Damadi, and X. Liu. “Privacy-Preserving Joint Data and Function Homomorphic Encryption for Cloud Software Services”. In: *IEEE Internet Things J.* 11.1 (2024), pp. 728–741.

[46] D. Hrestak and S. Picek. “Homomorphic encryption in the cloud”. In: *37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014, Opatija, Croatia, May 26-30, 2014*. IEEE, 2014, pp. 1400–1404.

[47] S. Goldwasser, S. Micali, and C. Rackoff. “The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)”. In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*. ACM, 1985, pp. 291–304.

[48] A. Fiat and A. Shamir. “How to Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *Advances in Cryptology - CRYPTO ’86, Santa Barbara, California, USA, 1986, Proceedings*. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194.

[49] M. Blum, P. Feldman, and S. Micali. “Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)”. In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. ACM, 1988, pp. 103–112.

[50] J. Groth. “Short Pairing-Based Non-interactive Zero-Knowledge Arguments”. In: *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 321–340.

[51] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. “From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again”. In: *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*. ACM, 2012, pp. 326–349.

[52] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza. “SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge”. In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. Vol. 8043. Lecture Notes in Computer Science. Springer, 2013, pp. 90–108.

[53] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. “Quadratic Span Programs and Succinct NIZKs without PCPs”. In: *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 626–645.

[54] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. “Bulletproofs: Short Proofs for Confidential Transactions and More”. In: *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 2018, pp. 315–334.

[55] D. Gabay, K. Akkaya, and M. Cebe. “Privacy-Preserving Authentication Scheme for Connected Electric Vehicles Using Blockchain and Zero Knowledge Proofs”. In: *IEEE Trans. Veh. Technol.* 69.6 (2020), pp. 5760–5772.

[56] Z. Wan, Y. Zhou, and K. Ren. “zk-AuthFeed: Protecting Data Feed to Smart Contracts With Authenticated Zero Knowledge Proof”. In: *IEEE Trans. Dependable Secur. Comput.* 20.2 (2023), pp. 1335–1347.

[57] G. N. Nithin, A. K. Pradhan, and G. Swain. “zkHealthChain - Blockchain Enabled Supply Chain in Healthcare Using Zero Knowledge”. In: *Internet of Things. Advances in Information and Communication Technology - 6th IFIP International Cross-Domain Conference, IFIPIoT 2023, Denton, TX, USA, November 2-3, 2023, Proceedings, Part II*. Vol. 684. IFIP Advances in Information and Communication Technology. Springer, 2023, pp. 133–148.

[58] S. Sahai, N. Singh, and P. Dayama. “Enabling Privacy and Traceability in Supply Chains using Blockchain and Zero Knowledge Proofs”. In: *IEEE International Conference on Blockchain, Blockchain 2020, Rhodes, Greece, November 2-6, 2020*. IEEE, 2020, pp. 134–143.

[59] J. Zhang, Z. Fang, Y. Zhang, and D. Song. “Zero Knowledge Proofs for Decision Tree Predictions and Accuracy”. In: *CCS ’20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. ACM, 2020, pp. 2039–2053.

[60] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang. “Mystique: Efficient Conversions for Zero-Knowledge Proofs with Applications to Machine Learning”. In: *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*. USENIX Association, 2021, pp. 501–518.

[61] S. Lee, H. Ko, J. Kim, and H. Oh. “vCNN: Verifiable Convolutional Neural Network Based on zk-SNARKs”. In: *IEEE Transactions on Dependable and Secure Computing* (2023), pp. 1–17.

[62] T. Liu, X. Xie, and Y. Zhang. “zkCNN: Zero Knowledge Proofs for Convolutional Neural Network Predictions and Accuracy”. In: *CCS ’21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 2021, pp. 2968–2985.

[63] J. Weng, J. Weng, G. Tang, A. Yang, M. Li, and J. Liu. “pvCNN: Privacy-Preserving and Verifiable Convolutional Neural Network Testing”. In: *IEEE Trans. Inf. Forensics Secur.* 18 (2023), pp. 2218–2233.

[64] C. Fang and E. Chang. “Differential privacy with δ-neighbourhood for spatial and dynamic datasets”. In: *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS ’14, Kyoto, Japan - June 03 - 06, 2014*. ACM, 2014, pp. 159–170.

[65] Y. Xiao and L. Xiong. “Protecting Locations with Differential Privacy under Temporal Correlations”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*. ACM, 2015, pp. 1298–1309.

[66] B. Hitaj, G. Ateniese, and F. Pérez-Cruz. “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 603–618.

[67] J. Gelhaar, T. Gürpinar, M. Henke, and B. Otto. "Towards a taxonomy of incentive mechanisms for data sharing in data ecosystems". In: *25th Pacific Asia Conference on Information Systems, PACIS 2021, Virtual Event / Dubai, UAE, July 12-14, 2021*. 2021, p. 121.

[68] N. J. King and V. T. Raja. "Protecting the privacy and security of sensitive customer data in the cloud". In: *Comput. Law Secur. Rev.* 28.3 (2012), pp. 308–319.

[69] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen. "Blockchain technology and its relationships to sustainable supply chain management". In: *Int. J. Prod. Res.* 57.7 (2019), pp. 2117–2135.

[70] T. Li, J. Vos, and Z. Erkin. "Decentralized Private Freight Declaration & Tracking with Data Validation". In: *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom 2022 Workshops, Pisa, Italy, March 21-25, 2022*. IEEE, 2022, pp. 267–272.

[71] R. Kromes, T. Li, M. Bouillion, T. E. Güler, V. van der Hulst, and Z. Erkin. "Fear of Missing Out: Constrained Trial of Blockchain in Supply Chain". In: *Sensors* 24.3 (2024), p. 986.

[72] H. Wu, Z. Li, B. King, Z. B. Miled, J. Wassick, and J. Tazelaar. "A Distributed Ledger for Supply Chain Physical Distribution Visibility". In: *Inf.* 8.4 (2017), p. 137.

[73] T. Bocek, B. Rodrigues, T. Strasser, and B. Stiller. "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain". In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, May 8-12, 2017*. IEEE, 2017, pp. 772–777.

[74] M. el Maouchi, O. Ersoy, and Z. Erkin. "DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*. ACM, 2019, pp. 364–373.

[75] P. Dutta, T.-M. Choi, S. Somani, and R. Butala. "Blockchain technology in supply chain operations: Applications, challenges and research opportunities". In: *Transportation Research Part E: Logistics and Transportation Review* 142 (2020), p. 102067. ISSN: 1366-5545.

[76] M. Rigaki and S. García. "A Survey of Privacy Attacks in Machine Learning". In: *ACM Comput. Surv.* 56.4 (2024), 101:1–101:34.

[77] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov. "Exploiting Unintended Feature Leakage in Collaborative Learning". In: *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 2019, pp. 691–706.

[78] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 3–18.

[79] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. “Learning Differentially Private Recurrent Language Models”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[80] E. D. Cristofaro. “An Overview of Privacy in Machine Learning”. In: *CoRR* abs/2005.08679 (2020). arXiv: 2005.08679.

2

DATA ANONYMIZATION

With the emerging of e-commerce, package theft is at a high level: It is reported that 1.7 million packages are stolen or lost every day in the U.S. in 2020, which costs $25 million every day for the lost packages and the service. Information leakage during transportation is an important reason for theft since thieves can identify which truck is the target that contains the valuable products. In this paper, we address the privacy and security issues in bin-packing, which is an algorithm used in delivery centers to determine which packages should be loaded together to a certain truck. Data such as the weight of the packages is needed when assigning items into trucks, which can be called bins. However, the information is sensitive and can be used to identify the contents in the package. To provide security and privacy during bin-packing, we propose two different privacy-preserving data publishing methods. Both approaches use differential privacy (DP) to hide the existence of any specific package to prevent it from being identified by malicious users. The first approach combines differential privacy with k-anonymity, and the other one applies clustering before differential privacy. Our extensive analyses and experimental results clearly show that our proposed approaches have better privacy guarantees, better efficiency, and better performance than the existing works that use either differential privacy or k-anonymity.

This chapter is a copy of the paper titled "Privacy-Preserving Bin-Packing with Differential Privacy" by Li, T., Erkin, Z., and Lagendijk, R. L. in *IEEE Open Journal of Signal Processing* vol. 3, pp. 94-106, 2022.

2.1. INTRODUCTION

Today, data plays an important role in our modern society. Many services such as transportation, supply chain logistics and healthcare are heavily dependent on data. On the one hand, more data improve the quality of services and even enable personalized ones. On the other hand, the collected data pose a serious threat in terms of privacy violations since the collected data are mostly privacy-sensitive or commercially valuable [1]. Considering container management systems for the transportation of goods, in the largest ports around the world, thousands of containers per day are being transported [2]. Trucks bring containers in and out, and while doing so, it is commercially important to use the container space as much as possible. To utilize the container space efficiently, different companies share the trucks to transport their products, and optimization algorithms are proposed to arrange the packages in containers [3, 4]. While doing so, it is also important to protect the commercially sensitive package data since such data can be obtained by malicious entities, resulting in the theft of certain products from the ports [5, 6]. As reported in a survey with 2000 respondents who have shopped online in the last 12 months [7], 43% of them reported a package stolen in 2020. Among them, 64% had more than one package stolen. Also, it is mentioned that information leakage is an important reason for truck theft, and thieves know which truck is the target that contains the valuable products [5]. In some cases, only the targeted products are stolen [8].

There are different processes during the transportation of packages that may leak information. In this paper, we address the privacy and security issues in bin-packing. The information of packages is needed when assigning items into bins. However, the information is sensitive and can be used to identify the contents in the package. Thieves can infer an iPhone or a MacBook in the package with a specific weight and volume since it always has the same weight and volume.

To protect data privacy and simultaneously use the optimization algorithm for better container management, the authors in [9] proposed a method to solve the bin-packing problem under privacy preservation. In that work, k-anonymity, which is a well-known technique for data anonymization [10], is used to publish anonymous container data. The authors use two k-anonymous algorithms: k-Optimize [11] and Flash [12], to publish data in a privacy-preserving manner. For every record in the dataset, there are $k-1$ same other records in the same dataset so that the record is indistinguishable. The authors use stochastic programming and robust optimization to address the uncertainty introduced by the k-anonymous published data that are fed to the optimizer. The authors clearly point out the trade-off between privacy guarantees and accuracy. However, the work completes computation in the order of minutes to hours for 25 or 50 items, which is with low efficiency. Meanwhile, the work is sensitive to the homogeneity attack since attackers can know the sensitive information if all the k tuples of quasi-identifiers share the same value in the sensitive attribute. Also, it is sensitive to the background knowledge attack since attackers can know the sensitive information based on some background knowledge. For example, there are k same packages, but the attacker knows the destination of the targeted package, and only one out of the k packages is heading to the targeted destination [13].

Machanavajjhala et al. [13] introduced l-diversity to mitigate attacks that exploit

similar sensitive attributes within the same k-anonymous group. A dataset satisfies l-diversity if each quasi-identifier group has at least l diverse sensitive values, adding complexity to an attacker's attempts to infer individual data. However, Li et al. [14] noted that l-diversity overlooks semantic relationships between sensitive values. They proposed t-closeness, which maintains that the distribution of sensitive attributes in each k-anonymous group is similar to that of the overall dataset, with t as the threshold for allowable distance. While t-closeness offers stronger privacy, it can be challenging to implement, as setting an appropriate t is difficult, and some sensitive attributes may inherently skew the distribution, limiting its practicality [15].

Besides approaches using k-anonymity, there are different privacy-preserving optimization methods, such as [16–18], in which only the optimization process is privacy-preserving. In these works, the optimizer knows the original information of packages and containers, which raises privacy risks in that the optimizer can be malicious by misusing the data or leaking information to other malicious users.

In this paper, we address the bin-packing problem as in [9]. In the adversary model, we assume that package data is anonymized and published in public. Then the published dataset is used for bin-packing optimization. The adversary aims to infer the content in the packages and locate the truck in which the target package is placed. They have access to the anonymized dataset, but they do not know the privacy parameters used for anonymization, such as k for k-anonymity or ϵ, δ for differential privacy.

To protect against the adversary, we focus on Differential Privacy (DP) [19, 20] instead of k-anonymity for two reasons: 1) to provide better privacy protection and 2) to achieve better efficiency in terms of run-time such that our proposals can be considered feasible in practice. We propose two algorithms based on DP:

- **Differential privacy with k-anonymity**. We first generate a lattice including all the possible generalization results of the input dataset with a given hierarchy, and then use the exponential mechanism [21] to output a specific generalization according to the utility. This method adds noise to the mapping function, which involves sampling, suppression and generalization selection. This method can reach a low value of ϵ for differential privacy and show low uncertainty based on the preset generalization hierarchy. However, the sampling and suppression result in only a proportion of data being processed.

- **Differential privacy with clustering**. We first cluster the data based on the number of occurrences, and then add Laplacian noise [20] to each cluster. This method directly adds noise to the data, resulting in a shorter run-time but introducing more noise, which has an impact on the performance.

Our security analysis and experimental results clearly show that our proposed methods provide better privacy and security guarantees than the previous work by comparing the probability of identifying the targeted package. The experiments show that the run-time of our proposed methods is significantly low, 0.1 seconds for 50 packages, while the previous work [9] needs several minutes or hours for anonymization. Also, the proposed methods achieve a comparable packing performance to the previous work [9].

The rest of the paper is organized as follows. In Section 2.2, we explain the preliminaries including differential privacy and k-anonymity. In Section 2.3, we present related

works about the existing privacy-preserving data publishing methods and optimization methods. Then Section 2.4 shows our two differential privacy-based data publishing methods followed by the security analysis in Section 2.5 and experimental results and analysis in Section 2.6. Finally, we give the conclusion and discussions in 2.7.

2

2.2. PRELIMINARIES

2.2.1. DIFFERENTIAL PRIVACY

Two datasets D and D' are neighbouring datasets if they only differ in one or zero rows of data, and an algorithm $\mathcal{A}$ satisfies ϵ-differential privacy (ϵ-DP) if and only if for neighbouring datasets D, D' and any set $O \subseteq \text{range}(\mathcal{A})$ [19, 20]:

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

However, the guarantee is so strong that it is very hard to be implemented, and it is excessive in many situations [22]. To make it more practical, parameter δ serves as a small error factor in the equation. $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy if:

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O] + \delta. \tag{2.2}$$

Based on the definition, the Laplace Mechanism and the Exponential Mechanism are two widely used mechanisms that satisfy differential privacy.

THE LAPLACE MECHANISM.

It is the most general mechanism for differential privacy, and it adds Laplace noise [20]. To add the noise, the mechanism applied Laplace distribution which is centred at zero with a scale parameter b:

$$Lap(x \mid \mu = 0,\ b) = \frac{1}{2b} \exp(-\frac{|x|}{b}). \tag{2.3}$$

We use $Lap(b)$ to denote density $Lap(x \mid \mu = 0,\ b)$. Then for the query $f : \mathcal{D}^N \rightarrow \mathbb{R}^k$, a randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if $\epsilon > 0$, k is the dimension of the dataset, and y_i is the noise added to dimension i:

$$\begin{aligned} \mathcal{A}(D,\ f,\ \epsilon) &= f(D) + (y_1\ \ldots\ y_k)\ , \\ \text{and} \quad y_i &\sim Lap\left(\frac{\Delta f}{\epsilon}\right). \end{aligned} \tag{2.4}$$

In Equation 2.4, Δf is the sensitivity for the query $f : \mathcal{D}^N \rightarrow \mathbb{R}^k$, and the l_1-*sensitivity* (Δf) is defined as:

$$\Delta f = \max_{X, X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} ||f(X) - f(X')||_1. \tag{2.5}$$

THE EXPONENTIAL MECHANISM.
The exponential mechanism [21] is a technique for designing algorithms with differential privacy. In the exponential mechanism, a utility function $u : \mathcal{D}^N \times \mathcal{R} \to \mathbb{R}$ is defined to access the utility of each element input $n \in \mathcal{R}$, where $\mathcal{D}$ is the domain and $\mathcal{R}$ is a range. Then a measure μ is used to assign a large probability of elements with a large utility.

With the utility function u, we calculate the *sensitivity* (Δu) of the utility function as:

$$\Delta u = \max_{n \in \mathcal{R}} \max_{X, X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} |u(X, n) - u(X', n)| , \tag{2.6}$$

and the output probability of the exponential mechanism is defined as:

$$\Pr[\mathcal{A}^{\epsilon'}_{u,\Delta u}(X) = t \in \mathcal{R}] = \frac{\exp(\epsilon' \cdot u(X, t)) \cdot \mu(t)}{\int_{\mathcal{R}} \exp(\epsilon' \cdot u(X, n)) \cdot \mu(n) dn} , \tag{2.7}$$

which satisfies ϵ-DP (where $\epsilon = 2\epsilon' \Delta u$).

2.2.2. BIN-PACKING PROBLEM

The *bin-packing* problem is an NP-hard optimization problem [23]. A real example is how to load packages into a minimum number of containers while avoiding overloading nor oversizing. The problem can be considered with different dimensions: *weight* and *volume* (*height*, *width* and *length*), which means that the problem can be with 1-D (*weight* or *volume*), 2-D (*weight* and *volume*) or 4-D (*weight*, *height*, *width* and *length*).

In this paper, we formulate the bin-packing problem as proposed in [24]. Considering 1-D bin-packing problem, for n items (or packages), we load them into the minimum number of bins (or containers). w_j is the weight of item $j \in N$, where $N = \{0, 1, 2, ..., n\}$, and all the bins have capacity c. We define the decision variables y_i and $x_{i,j}$ as follows:

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used,} \\ 0 & \text{if bin } i \text{ is not used,} \end{cases} \tag{2.8}$$

$$x_{i,j} = \begin{cases} 1 & \text{if item } j \text{ is loaded in bin } i, \\ 0 & \text{if item } j \text{ is not loaded in bin } i. \end{cases} \tag{2.9}$$

Given y_i and $x_{i,j}$, as shown in Equation 2.8 and 2.9, the formulation of the 1-D bin-packing problem is:

$$\min \sum_{i \in N} y_i , \tag{2.10}$$

$$\text{s.t.} \sum_{j \in N} w_j x_{i,j} \leq c y_j \qquad \forall i \in N, \tag{2.11}$$

$$\sum_{i \in N} x_{i,j} = 1 \qquad \forall i, j \in N. \tag{2.12}$$

In Equation 2.10, the objective is to minimize the number of bins, and the two constraints ensure that every bin is not overloaded and one item can only be loaded into one bin.

2.2.3. THE FRAMEWORK FOR BIN-PACKING

Figure 2.1 shows the framework used in this paper. The framework was proposed in [9], including two modules: the data publishing module and the optimizer module. In the data publishing module, we apply anonymization methods to the private dataset and publish the differentially private (DP) dataset to the public. Then the optimizer module gets data from the public and applies optimization to solve the bin-packing problem using the anonymous data. The whole framework is privacy-preserving since the optimization is based on anonymous data.

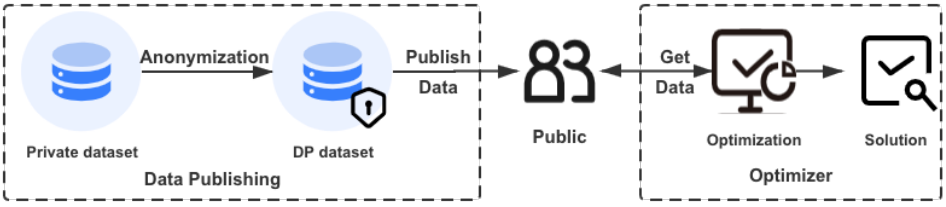


Figure 2.1: Framework overview.

2.3. RELATED WORK

Data anonymization is a technique to achieve privacy protection in data mining. The idea is to analyze data without revealing users' sensitive information [25]. Among many approaches, data perturbation methods [26, 27] attracted significant attention in recent years. By applying data perturbation, a certain amount of noise is added to the raw dataset to achieve data anonymization. The noise decreases the utility of the dataset while preserving users' privacy by adding uncertainty to the dataset. Two widely used methods are k-anonymity [10] and differential privacy [19, 20], which are based on data generalization and adding random noise.

The concept of k-anonymity was introduced by Samarati and Sweeney in 1998 [10]. A dataset is k-anonymous if, for each individual in the dataset, there are at least $k-1$ other individuals which show the same value. There are a variety of k-anonymous algorithms for data anonymization. For example, Datafly [28] is a heuristic k-anonymous algorithm, which generalizes the quasi-identifiers showing the most distinct values. Mondrian [29] is another modern k-anonymous algorithm proposed by LeFevre et al. By using *kd-tree*, Mondrian splits the dataset and reconstructs it with equivalence classes whose size is at least k. Also, Emam et al. [30] proposed OLA, which achieves k-anonymity by using a pre-defined *generalization hierarchy* with generalization rules for each attribute.

In 2019, Hoogervorst et al. [9] applied k-anonymity to the bin-packing problem to publish the weights of packages. The authors used full domain generalization and partition-based single-dimensional recoding to generalize the data. Also, two k-anonymous algorithms: k-Optimize [11] and Flash [12], are evaluated. However, k-anonymity is sensitive to the homogeneity attack and the background knowledge attack [13]. Meanwhile, k-anonymity brings uncertainty for the optimization, so the authors also applied stochastic programming and robust optimization to improve the performance of bin-

packing. As far as we know, this is the only literature which applied anonymization techniques to the input data for bin-packing instead of proposing a privacy-preserving optimizer.

Different from k-anonymity, differential privacy aims to hide the existence of any single row of data in the dataset. Differential privacy can be applied to either add noise to the output of a certain query (such as the optimization in [16]) or add noise to the dataset [26, 31–34]. The work of [33] and [34] consider the trajectory data release using differential privacy. Hyukki Lee and Yon Dohn Chung [26] released the medical microdata in a differentially private way. They applied generalization, suppression and insertion to add noise to the data. Moreover, they used the exponential mechanism to maximize the utility of the output dataset. The CASTLEGUARD [32] applied the Laplace mechanism to the numerical data to get a differentially private dataset, but the output is noisy and sparse with a low value of ϵ. Also, Holohan et al. [31] applied k-anonymity to part of the attributes and differential privacy to the rest. Similar to the work of CASTLEGUARD, they also applied the Laplace mechanism to the numerical data. Besides, they gave a confidence interval for the perturbation. We used this method in Section 2.4.2.

Overall, from the literature, there are two main techniques for data anonymization: k-anonymity and differential privacy. However, k-anonymity based approaches are sensitive to background knowledge attacks and need a long run time (several minutes or hours) to find the optimal. Meanwhile, existing differential privacy based methods introduce large noise to the dataset for bin-packing problems, which can influence the performance. To achieve better efficiency, better privacy guarantees (compared to k-anonymity solutions) and better performance for bin-packing (compared to the existing differential privacy solutions), we propose two different approaches by (1) combining the use of k-anonymity and differential privacy and (2) applying clustering with differential privacy.

2.4. DATA ANONYMIZATION USING DIFFERENTIAL PRIVACY

In this section, we present two data anonymization methods based on differential privacy with different approaches and strengths. The first method combines differential privacy and k-anonymity using preset generalization hierarchy and the differentially private node selection method, which shows better privacy guarantee but lower efficiency. The second method adds Laplace noise to the data in each cluster, which works more efficiently since all items are considered each time, but it is with a lower privacy guarantee.

2.4.1. DIFFERENTIAL PRIVACY WITH K-ANONYMITY

This subsection shows a data anonymization method that combines differential privacy with k-anonymity. We firstly generalize data based on the full-domain generalization method and construct a differential private mapping function based on a k-anonymous algorithm OLA [30]. Then we prove that the new k-anonymization method satisfies differential privacy.

FULL-DOMAIN GENERALIZATION

Full-domain generalization [35] is a widely used method for recoding [10]. For different quasi-identifier attributes Q_i, a generalization function ϕ_i is defined as $\phi_i : D_{Q_i} \rightarrow D_{G_i}$, and $D_{Q_i} \leq_D D_{G_i}$, which means that D_{G_i} is generalized from D_{Q_i}. D_{Q_i} is the original dataset and D_{G_i} is the generalized dataset. For each value $q \in D_{Q_i}$, ϕ_i maps it to $g \in D_{G_i}$, and we can get that $g \in \gamma^+(q)$ (which means that g is a generalization of q), or $g = q$. In a full-domain generalization, all values q for all attributes Q_i are replaced by $\phi_i(q)$.

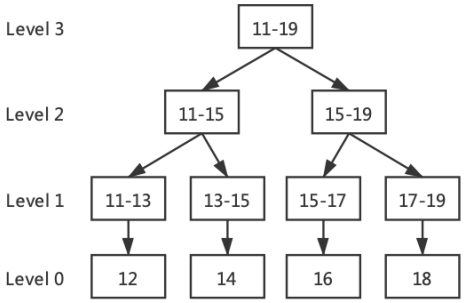


Figure 2.2: An example full-domain generalization.

In Figure 2.2, we give an example of the possible generalization of four values {12, 14, 16, 18}. For the value **12**, we can generalize it into "11-13" or "11-15" or "11-19" or remain as "12". By generalization, we add some uncertainty to the data, which can decrease the utility but better protect privacy. The generalization is independent of data distribution, and instead, it is determined by the attribute. Also, with the generalization, the generalized value of different inputs may be the same, such as "12" and "14" may both output "11-15". The full-domain generalization method is used for k-anonymity since it reduces utility (with more generalization) to achieve k-anonymity. However, the generalization can only be used for 1-D bin-packing problems, since it is not possible to generalize a 2-D tuple in a same way. In this paper, we combined OLA and differential privacy to show a solution.

LATTICE-BASED STRUCTURE

Firstly, we define different levels to show how much an attribute is generalized. As shown in Figure 2.2, level zero means that no generalization is applied, and level three means that the data is fully generalized. Based on the definition, we use a lattice-based structure to decide how many generalizations should be applied when using the full-domain generalization. The structure is proposed in a k-anonymous algorithm OLA [30]. Figure 2.3 gives an example when there is only one attribute, and $< 0 >, < 1 >, < 2 >, < 3 >$ are all the nodes in the lattice.

Then we expand it to be with two attributes as shown in Figure 2.4. Each node indicates a different generalization of an attribute. The lattice becomes larger with a deeper full-domain generalization hierarchy or more attributes.

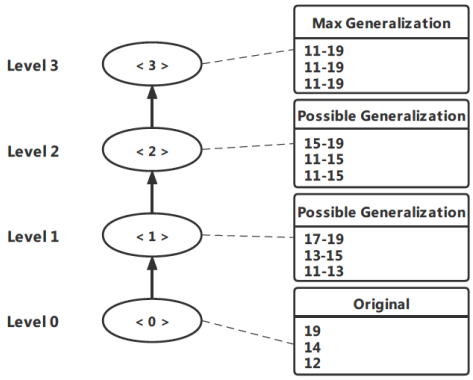


Figure 2.3: An example lattice with level 2 for one attribute.

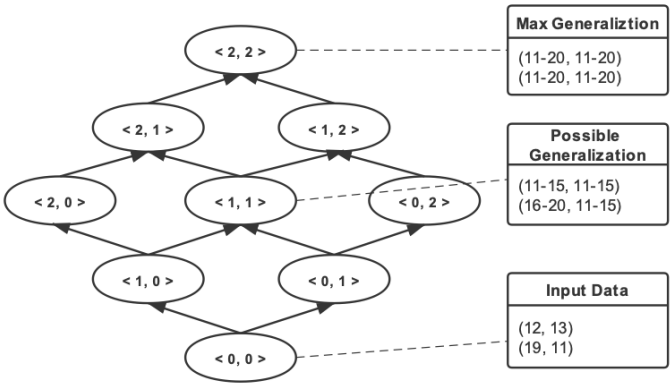


Figure 2.4: An example lattice with level 2 for two attributes.

APPLICATION OF DIFFERENTIAL PRIVACY

Li et al. [22] give the idea of differential privacy under-sampling (β,ϵ,δ)-DPS where β is the sampling factor, ϵ is the privacy budget, and δ is the small error factor for differential privacy. The sampling means that every record is only with probability β being selected from the original dataset, otherwise it is removed. For an algorithm $\mathcal{A}$, if $\mathcal{A}^\beta$ is ϵ-DP, $\mathcal{A}$ satisfies (β,ϵ,δ)-DPS. $\mathcal{A}^\beta$ means that the dataset is firstly sampled with probability β, and a smaller β results in a smaller ϵ. The same paper also proves that if the mapping function $\mathcal{A}_m$ of a k-anonymization algorithm satisfies ϵ_1-DP, the k-anonymization algorithm satisfies (β,ϵ,δ)-DPS where

$$\epsilon \geq -\ln(1-\beta)+\epsilon_1, \tag{2.13}$$

$$\begin{aligned} \delta &= d(k,\beta,\epsilon-\epsilon_1) = \max_{n:n\geq\lceil\frac{k}{\gamma}\rceil} \sum_{j>\gamma n}^{n} f(j;n,\beta), \\ \gamma &= \frac{(e^{\epsilon-\epsilon_1}-1+\beta)}{e^{\epsilon-\epsilon_1}}, \end{aligned} \tag{2.14}$$

in which $f(j;n,\beta)$ returns the probability of achieving j successes in n trials and the probability of a successful trial is β.

Algorithm 1: Differential privacy with k-anonymity

Input: Input dataset D_{in}, privacy budget ϵ_1
Output: Differentially private dataset D_{out}
1: Apply the β sampling to D_{in}, and get D'_{in}
2: Construct the lattice generalizations for attributes of D'_{in}
3: Calculate the utility of each node by Equation 2.15
4: Compute the probability for every node to be selected as the output (using the exponential mechanism with ϵ_1)
5: Randomly pick a node n_i according to the probability
6: Generalize the dataset D_{out} for n_i
7: Suppress the records which do not satisfy k-anonymity
8: **return** D_{out}
9: *Note: sensitivity Δu can be calculated anywhere and the algorithm satisfies $2\epsilon_1\Delta u$-DP.*

Based on the definition of ϵ-DP k-anonymization algorithm, we present Algorithm 1. We firstly apply the sampling, which means that every record is with a probability β being selected from the original dataset. In the second step, we generate the lattice based on the generalization hierarchies. Then in step 3 in Algorithm 1, we calculate the utility of each node using the utility function in Equation 2.15 with consideration of privacy and information loss. Intuitively, we want the algorithm with a higher privacy guarantee and lower information loss. For the privacy part $sup(D,n)$, we consider k-anonymity in terms of the proportion of the suppressed data. For the information loss part $gen(D,n)$, we consider how many levels have been generalized.

$$u(D,n) = sup(D,n) \cdot gen(D,n)\ , \tag{2.15}$$

where

$$sup(D,n) = \frac{|D_{k-anonymity}|}{|D_{raw}|} \in [0,1]\ , \tag{2.16}$$

$$gen(D,n) = 1 - \frac{1}{N_A}\sum_{i=1}^{N_A}\frac{n_{A_i}}{|FDG_{A_i}|} \in [0,1]\ . \tag{2.17}$$

Equation 2.15 shows the trade-off between the information loss ($gen(D,n)$) and the privacy concern ($sup(D,n)$). Ideally, the output node is with the highest utility value. In Equation 2.16, we choose the remaining proportion of the dataset to ensure that a higher value of $sup(D,n)$ represents better privacy guarantee. In Equation 2.17, N_A is the number of attributes, n_{A_i} is the generalized level, and $|FDG_{A_i}|$ is the fully generalized level.

Based on the utility function and Equation 2.7, we can calculate the output probability of the exponential mechanism in step 4 in Algorithm 1 as shown in Equation 2.18. With the output probability for each node, a node is selected as the output node.

$$\Pr[\mathcal{A}^{\epsilon'}_{u,\Delta u}(D) = t \in \mathcal{R}] = \frac{\exp(\epsilon' \cdot u(D,t)) \cdot \mu(t)}{\int_{\mathcal{R}} \exp(\epsilon' \cdot u(D,n)) \cdot \mu(n) dn}. \tag{2.18}$$

Equation 2.18 satisfies ϵ-differential privacy (where $\epsilon = 2\epsilon'\Delta u$), and the *sensitivity* (Δu) of the utility function is:

$$\Delta u = \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |u(D,n) - u(D',n)|. \tag{2.19}$$

The sensitivity shows the maximum change of the value of the utility function if we change only one row of data in the dataset. For the utility function in Equation 2.15,

$$\begin{aligned} \Delta u &= \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |u(D,n) - u(D',n)| \\ &= \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |sup(D,n) - sup(D',n)| \\ &\quad \cdot gen(D,n) \\ &\leq |sup(D,n) - \left(sup(D,n) + \frac{k}{|D|} \right)| = \frac{k}{|D|}. \end{aligned} \tag{2.20}$$

With the equations, the mapping function satisfies ϵ_1-DP with the exponential mechanism, so Algorithm 1 satisfies $(\beta, \epsilon, \delta)$-DPS as in Equation 2.13.

DISCUSSIONS

The proposed method can be expanded to be used for different data anonymization tasks with both categorical and numerical data. Also, the method can be applied to datasets with different dimensions. The proposed method can be used as a general scheme, but we only consider it for the bin-packing use case in this paper.

Meanwhile, the complexity of the approach is influenced by the number of attributes and records of a dataset. When the number of attributes increases, the lattice will increase exponentially, resulting in a long run time.

2.4.2. DIFFERENTIAL PRIVACY WITH CLUSTERING

This subsection shows another anonymization method in which we adopt clustering before applying the Laplace mechanism, as shown in Algorithm 2.

Section 2.3 shows that we can add noise to the raw dataset to satisfy differential privacy, which is used in [31] and [32]. In the bin-packing problem, all the attributes are numerical, so we can add Laplace noise to the value of each attribute (v_i) as:

$$v'_i = v_i + Lap\left(\frac{\Delta v_i}{\epsilon}\right), \tag{2.21}$$

where $\Delta v_i = \max(v_i) - \min(v_i)$. Here the sensitivity is defined as the difference between the largest and lowest possible weight. If the weight is anonymous among this range, it is anonymous among all the packages.

By adding Laplace noise, the output v'_i satisfies ϵ-DP. However, sometimes customers do not want to change the value of their products. For example, the *weight* is 10kg and

the *volume* is 1m^3, and we publish it as 12kg and 0.8m^3. However, for express or logistics, the price is based on weight and volume. It can cause a problem if the differentially private value is not close to the accurate one. Considering this problem, the published dataset is only used to optimize the bin-packing problem, such as how to load packages into a minimum number of containers. Also, we introduce a confidence $c \in [0,1]$, and Holohan et al. [31] show that the probability

$$\mathcal{P}(v_i \in [v_i' - r_c, v_i' + r_c]) = c\,, \tag{2.22}$$

$$\text{where } r_c = -\frac{\Delta v_i}{\epsilon} \ln(1-c). \tag{2.23}$$

By applying that, we can publish an interval instead of a single value. With the confidence c, we can control the probability of whether the accurate value is in the interval.

Equation 2.21 shows that the noise is influenced by outliers, such as the extremely large or heavy packages. In order to reduce the influence of outliers, we adopt clustering before applying differential privacy.

Here the clustering is based on the proportion of occurrence. For example, we can divide the input dataset into five parts by 5%, 30%, 30%, 30% and 5%. By applying the clustering, we can ease the problem of outliers, but it only satisfies differential privacy within each cluster. Most data are anonymous among the 30% records, which show similar weights or volumes.

To some extent, the clustering method extends the restriction of differential privacy. The proposed method anonymizes any single record among its cluster instead of the whole dataset. It is a trade-off between utility and privacy. There are thousands of packages in real use, and being anonymous among its cluster, which is with hundreds of packages, is still secure, as shown in Section 2.5.

In Algorithm 2, the sensitivity is calculated for each cluster with complexity $O(n_c)$, and the noise is added to the weight of each package with complexity is $O(n_p)$, so the complexity for Algorithm 2 is $O(n_c) + O(n_p)$ where n_c is the number of clusters and n_p is the number of packages.

Also, the differential privacy with clustering method can be expanded to different data anonymization tasks, but it is restrictive since only numerical data with low dimensions can be considered. With high dimensions, there are a large number of clusters, and only a few records are in each cluster, which makes it infeasible. In this paper, we consider the bin-packing problem, which is a suitable use case for the approach.

2.5. SECURITY ANALYSIS

This section analyzes and compares the privacy guarantees provided by the k-anonymity method in [9], the DP with k-anonymity method and the DP with clustering method in Section 2.4. As mentioned in Section 2.3, the work of [9] is the only literature which considered privacy in bin-packing. There are other works which applied differential privacy for anonymization, such as [31], but the privacy guarantee is the same as our proposed approaches since differential privacy is applied to all of them. For that work, the performance for bin-packing is further compared in Section 2.6.

Algorithm 2: Differential privacy with clustering

Input: Input dataset for de-identification D_{in}
Output: Output dataset D_{out}
1: Sort D_{in}
2: Apply clustering to D_{in} based on the proportion of occurrence
3: Calculate Δv_i for each cluster
4: Calculate v_i' by adding Laplace noise to each cluster using Equation 2.21
5: Calculate the interval of each v_i' using Equation 2.22 and 2.23
6: Get D_{out} by combining the output of each cluster
7: **return** D_{out}

In this paper, we assume that the adversary knows the accurate information of one package and wants to identify this package from the anonymous output. If the adversary can identify the package, they know which container the package is loaded to, and can thus track the package. To quantify how well privacy is protected in this scenario, we compare the probability that an adversary can identify the correct package from the output dataset.

In the work of [9], only k-anonymity is considered. Each row of data occurs at least k times in the output dataset. We can calculate the probability of identifying the same package from the output dataset given the information of the target package, as shown in Equation 2.24. In the scenario, the adversary knows the original weight a_i (such as $a_i = 12$) and wants to identify which b_i is its output. The adversary first finds all possible b_i which show the correct generalization for a_i (such as $[10, 15]$). Based on the definition of k-anonymity, there are at least k possible b_i showing the same generalization $[10, 15]$, so the probability is at most $1/k$.

$$\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}] \leq \frac{1}{k}. \tag{2.24}$$

In the differential privacy with k-anonymity method in Section 2.4.1, we add uncertainty to the dataset using sampling, generalization and suppression. Compared to the work of [9], this approach applies β random sampling and differentially private mapping, which achieves $(\beta, \epsilon, \delta)$-DP. On the one hand, in the output dataset, every single row of data is hidden in a crowd. Based on the definition of differential privacy, the probability of outputting a specific record changes less than e^{ϵ} if we change any record in the input dataset. On the other hand, this approach applies k-anonymity with sampling and differentially private mapping. The β sampling adds more uncertainty in that the adversary does not know whether the target package is in the input dataset or not. Even if the adversary gets all the possible b_i, they do not know whether the correct data is included. Equation 2.25 shows the new probability equation and $0 < \beta < 1$.

$$\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}] \leq \beta\frac{1}{k}. \tag{2.25}$$

Besides, the differentially private mapping function provides stronger privacy guarantees. In k-anonymous algorithms, the mapping is usually based on the existence of a

few values [22]. For example, if the dataset is $\{1,2,3,5,7,9\}$ and $k=3$, one of the possible generalizations is $\{[1,3],\ [5,9]\}$, which shows the existence of "1, 3, 5, 9" in the input dataset. The differentially private mapping does not overly depend on any single record in the input dataset. Each possible generalization can be chosen as the final output concerning their probability from the exponential mechanism [22]. As a result, the mapping function enhances the privacy guarantee, but it cannot be shown in Equation 2.25.

In the differential privacy with clustering method in Section 2.4.2, we add Laplace noise to each cluster to hide the existence of any single row of data in each cluster. For example, if we have a dataset D: $\{a_0, a_1, ..., a_5\}$ with two clusters C_1: $\{a_0, a_1, a_2\}$ and C_2: $\{a_3, a_4, a_5\}$. The output dataset is D':

$$\begin{aligned}\{&a_0 + \text{Lap}(\delta_1/\epsilon),\ a_1 + \text{Lap}(\delta_1/\epsilon),\ a_2 + \text{Lap}(\delta_1/\epsilon),\\ &a_3 + \text{Lap}(\delta_2/\epsilon),\ a_4 + \text{Lap}(\delta_2/\epsilon),\ a_5 + \text{Lap}(\delta_2/\epsilon)\}\end{aligned} \tag{2.26}$$

where the sensitivity

$$\delta_i = \max_{a_x, a_y \in C_i} |a_x - a_y|. \tag{2.27}$$

Assume that the adversary knows $x_{target} = x_2$ from D and the output dataset D'. They want to identify x_2 from D', so they calculates the difference between the accurate data and the output data, getting:

$$\begin{aligned}\{&\Delta a_0 + \text{Lap}(\delta_1/\epsilon),\ \Delta a_1 + \text{Lap}(\delta_1/\epsilon),\ \Delta a_2 + \text{Lap}(\delta_1/\epsilon),\\ &\Delta a_3 + \text{Lap}(\delta_2/\epsilon),\ \Delta a_4 + \text{Lap}(\delta_2/\epsilon),\ \Delta a_5 + \text{Lap}(\delta_2/\epsilon)\}.\end{aligned} \tag{2.28}$$

where $\Delta a_i = a_{target} - a_i$.

If the adversary infers that the noise is generated by the Laplace mechanism, they know the probability density function for Laplace distribution:

$$f(x|\mu, b) = \frac{1}{2b} \exp(-\frac{|x-\mu|}{b}). \tag{2.29}$$

Based on the probability density function, the adversary can get the probability equation:

$$\begin{aligned}&\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}]\\ &= \frac{f(b_{target} - a_{target}\ |\mu = 0, b = \delta_1/\epsilon)}{\sum_{b_i \in D_{out}} f(b_i - a_{target}\ |\mu = 0, b = \delta_i/\epsilon)}\\ &= \frac{f(\text{Lap}(\delta_1/\epsilon)\ |\mu = 0, b = \delta_1/\epsilon)}{\sum_{b_i \in D_{out}} f(\Delta a_i + \text{Lap}(\delta_i/\epsilon)\ |\mu = 0, b = \delta_i/\epsilon)}.\end{aligned} \tag{2.30}$$

However, in Equation 2.30, the adversary cannot get access to the value of δ_i and ϵ, so they cannot get the result of the probability. Meanwhile, Equation 2.30 shows that $\text{Lap}(\delta_i/\epsilon)$ influences the output probability. With a high sensitivity δ_i or a low ϵ, the variance of the Laplace noise is large. The result of the $\text{Lap}(\delta_i/\epsilon)$ counts equally or more than Δa_i, which can hide any record in the cluster.

In conclusion, both our proposed methods show better privacy guarantees, which can lower the probability that a potential attacker identifies targeted packages from the group.

2.6. EXPERIMENTAL EVALUATION

This section shows the experimental evaluation of the proposed methods in Section 2.4. We use Python to implement both methods on a laptop with Windows 10 Pro, Intel Core i7-10710U CPU and 16.0 GB RAM. We use Google Or-Tools [36] for optimization. We have compared the performance of our proposed methods to the existing methods using k-anonymity [9] or differential privacy [31] with seven different synthetic datasets.

BPPLIB [37] has given different benchmarks for bin-packing, such as Falkenauer [38], Scholl [39], and the Randomly Generated Instances [40]. Among them, the datasets are generated following the uniform distribution with a different number of items (n), capacity (c), minimum (l) and maximum (u) values. These datasets have a variety of combinations of these four factors (n, c, l, u) to test the performance of the optimization algorithms for bin-packing. However, this paper focuses on evaluating the proposed anonymization algorithms in terms of the performance for bin-packing, feasibility and run-time, instead of assessing the optimization methods. In the experiments, we consider more distributions such as normal distributions and uniform distributions, but fewer combinations of the four factors. To properly evaluate both proposed approaches, different instance settings are applied, and the settings are further introduced in Tables 2.1 and 2.2.

This section first shows the optimization methods for bin-packing and introduces the factors to evaluate the performance. After that, we demonstrate the performance of the proposed methods, in which the instance setting and performance analysis are included. Finally, we compare the performance of our proposed approaches to the existing works.

2.6.1. OPTIMIZATION METHODS

Equation 2.10 shows how the standard optimization works, and the optimization result is the number of bins needed to load all the items. Note that the bin-packing problem is computationally NP-hard. The optimization method is how the problem is solved, so the optimization methods influence the global performance in terms of run time and whether the optimal is found. There are different optimization methods for bin-packing, such as the work of [3, 4]. In this paper, the performance of the optimization methods is not our focus, and we choose a widely used optimization tool (Google Or-Tools) in all the experiments and set a time limit (1 minute) for optimization.

In the experiments, we can apply the upper bound or the mean value to the standard optimization for the anonymous data. With the upper bound, the optimization for Algorithm 1 is ensured to be feasible for the containers. The optimization for Algorithm 2 is feasible with at least the probability of the confidence c in Equation 2.22. With the upper bound, the solution is feasible to the containers, but it can also lead to container space waste since the weights can be largely overestimated. With the mean value of the interval, we can avoid the overestimated weights. However, it also increases the risk that the container is overloaded, making the solution infeasible to the constraints.

2.6.2. PERFORMANCE METRICS

To evaluate the performance of the proposed methods, we introduce different factors. Also, to mitigate the influence of the randomness for differential privacy, every experi-

ment is carried out ten times, and the average is used as the result.

Objective ratio (o/o_n). o is the optimization result using the output data from the proposed algorithms, and o_n is the optimization result using the original data. The optimal objective ratio is 1, since a ratio larger than 1 means more bins are used, and a ratio less than 1 means some bins must have violated the restrictions.

Feasibility f. For each bin b_i, the optimization result using the anonymous data can violate the constraints in Equation 2.10. For example, two anonymous items whose weights show as $\{11.2, 13.6\}$ are loaded to a container with capacity = 25, but the accurate weights of these items are $\{12, 15\}$, which violates the constraint. To evaluate how often the violation happens, we use the feasibility value f to represent the proportion of the bins that satisfies all the constraints using the accurate data. If $B = \{b_0, b_1 ..., b_m\}$ is the optimization result that uses m bins to load all the items and $D(b_i)$ is the accurate weights of the items in bin i, then

$$f = \frac{\sum_{b_i \in B} g(b_i,\, D(b_i))}{|B|}, \tag{2.31}$$

$$\text{where } g(b_i,\, D(b_i)) = \begin{cases} 0 & \text{if bin } i \text{ violates constraints,} \\ 1 & \text{otherwise.} \end{cases} \tag{2.32}$$

Anonymization time t_a. The run-time to run the proposed methods. We use the anonymization time to evaluate the efficiency of the methods.

Suppression rate. We introduce the suppression rate to evaluate how much data is suppressed in the differential privacy with k-anonymity method.

2.6.3. PERFORMANCE OF DIFFERENTIAL PRIVACY WITH K-ANONYMITY

INSTANCE SETTINGS.

Seven different instance settings are evaluated, as shown in Table 2.1. Similar to the benchmarks in BPPLIB, we consider uniform distribution in instances I to IV with the same distribution as the instances used in the work of [9]. Setting I and II have different numbers of medium and large items with uniform distribution (U). Similarly, we increase the capacity from 500 to 2500 to evaluate the small items in setting III and IV. Also, we add the normal distribution to consider a different distribution. Instance VI is with a combination of two uniformly distributed sub-sets, which is also with the same distribution as used in [9]. It is with 25% large items and 75% small items. Instance VII is generated by [39] with more items (200) and the optimization is hard to be solved. This instance is supposed to show how well different algorithms work on a larger dataset.

In Table 2.1, c is the capacity of the bins; the weights of all the items are in the range of $[l \cdot c,\ u \cdot c]$; n is the number of items. Due to the suppression by k-anonymity, the number of items is larger than the settings for the clustering method in Table 2.2. In the settings, 'L' means large items, 'U' means uniform distribution, and 'N' means normal distribution.

PARAMETER SETTINGS.

In the $(\beta, \epsilon, \delta)$-DP with k-anonymity method, Equation 2.13 and 2.20 show that:

$$\epsilon \geq -\ln(1-\beta) + \epsilon_1 = -\ln(1-\beta) + 2\epsilon' \cdot \frac{k}{|D|}, \tag{2.33}$$

Table 2.1: Experimental settings for DP with k-anonymity.

setting	c	l	u	n	distribution
I. 50/L/U	500	0.25	0.75	50	100%: U(125,375)
II. 80/L/U	500	0.25	0.75	80	100%: U(125,375)
III. 50/S/U	2500	0.05	0.15	50	100%: U(125,375)
IV. 80/S/U	2500	0.05	0.15	80	100%: U(125,375)
V. 50/L/N	500	0	1.00	50	100%: N(250,100)
VI. 50/L/Un	500	0.25	0.75	50	75%: U(125,250), 25%: U(250,375)
VII. 200/L	100000	0.2	0.35	200	As used in [39]

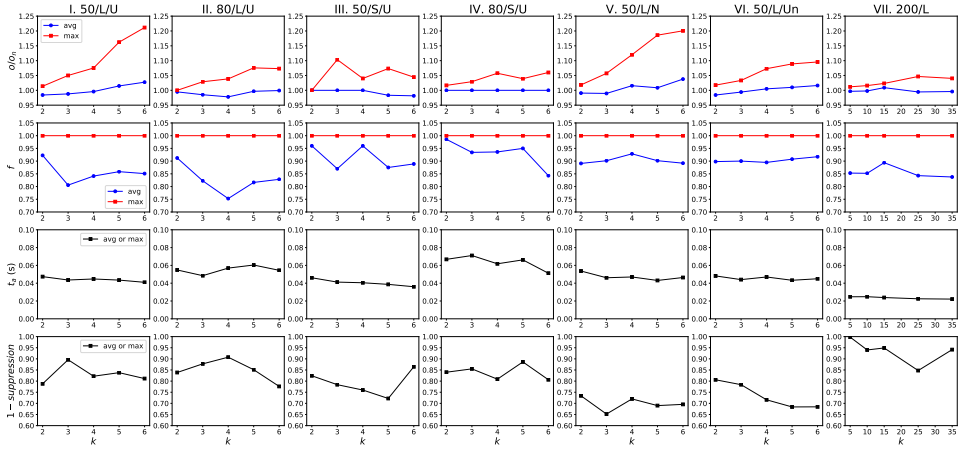


Figure 2.5: The performance of the differential privacy with k-anonymity method using the average or the upper bound of intervals (with $\epsilon' = 3$). The x-axis is k, and the y-axis is: the objective ratio o/o_n, the feasibility f, the anonymization time t_a (s), and the proportion of the remaining data $1 - suppression$.

where β is the sampling rate and $\epsilon_1 = 2\epsilon'\Delta u$ is for the ϵ_1-DP mapping function. In the evaluation, we assume that the instances in Table 2.1 are **after** the β sampling. We choose the number of $k \in [2,6]$ as the independent variable to evaluate the performance since the value of ϵ (in Equation 2.33) and δ (in Equation 2.14) are both dependent on k. Meanwhile, we set $\epsilon' = 3$ to achieve a relatively small value of ϵ. For example, with $k = 4$, $|D| = 40$, $\beta = 0.7$, we can get $\epsilon \approx 1.8$. Due to the randomness of differential privacy, we carry out every experiment ten times and use the average value for evaluation. Also, it is time-consuming to get the optimal solution for an optimization problem, so we set a time limit of 1 minute for the standard optimization.

PERFORMANCE ANALYSIS.

Figure 2.5 shows the performance of the differential privacy with k-anonymity method. We use both the average (avg) and the upper bound (max) of the output intervals as the input to the standard optimizer. The average performs better than the upper bound in terms of objective ratio at the cost of feasibility. The weights of items are overestimated

with the upper bound, leading to a larger objective ratio ranging from 1.0 to 1.2 (k = 6). For the same reason, the optimization results using the upper bound always satisfy all the constraints. On the contrary, the average weights are closer to the real, but the weights can be underestimated, resulting in overloaded bins.

For most settings with the upper bound, the objective ratio increases with a higher value of k. With a larger k, the exponential mechanism is more likely to choose a node with more generalization to keep a low suppression proportion. With more generalization, the upper bound is more overestimated, which increases the objective ratio. Meanwhile, the objective ratio is more close to 1 with a larger dataset. For instance setting VII, the objective ratio is close to 1 even using the upper bound.

For the flexibility, it is not always equal to 1 if the average bound is applied. The probability of violation is around 10% to 20%. To mitigate this problem, we can set the capacity a bit smaller than the real capacity. Also, in practice, we can drop some products to satisfy the constraints.

The suppression rates are different among different distributions. For uniform distributions, the suppression is around 10% to 25%, which means that only a small proportion of data are suppressed. For the normal distribution in setting V, the suppression rises to around 30% since weights are sparse for the large/small items. For a similar reason, values are sparse for the large items with the nonuniform distribution, resulting in a higher suppression (20% to 30%). When the number of items increases, the suppression rate is only with around 10% even when $k = 35$, which shows its advantages in large datasets.

The suppression also introduces a problem that not all the items are considered for bin-packing. To deal with that, there are three different approaches:

- Keep the items into the next pool and wait for k items with the same range for k-anonymity.
- Apply differential privacy directly or apply Algorithm 2 to the suppressed data.
- Consider more about the suppression in the utility function, so the utility function can guarantee that the output is with a low suppression.

Both the low suppression rates and the low objective ratio show that the proposed utility function works well. Also, the run-time for the anonymization algorithm is less than 0.1 seconds to output an anonymous dataset.

Equation 2.33 shows that a smaller k means a smaller ϵ, but this is with limits. When we calculate δ using Equation 2.14, if k is small, the value of δ is large. Dwork et al. [41] show that δ should be smaller than $1/|D|$, where $|D|$ is the number of records in the dataset. The value of δ is large with a small-scale dataset and a small k, but δ can satisfy it with a large dataset and a suitable k. For example, if $|D| = 1000, \beta = 0.7, k = 40, \epsilon_1 = 1$, we can get $\delta \leq 6.8 \times 10^{-4} < 1/|D|$. In real use, there are thousands of items being loaded everyday. We can select the minimum k, which satisfies the restriction.

2.6.4. PERFORMANCE OF DIFFERENTIAL PRIVACY WITH CLUSTERING

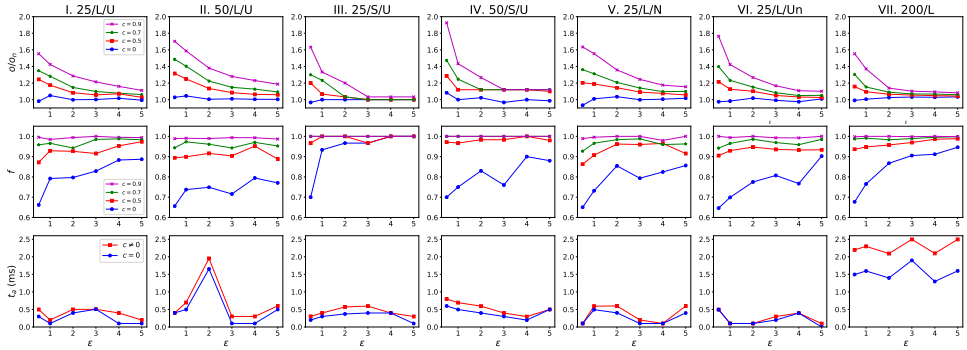


Figure 2.6: The performance of the differential privacy with clustering method with different confidence factor c. The x-axis is ϵ, and the y-axis is: the objective ratio o/o_n, the feasibility f, and the anonymization time t_a (ms).

EXPERIMENTAL SETTINGS.

Table 2.2 shows the instance setting, which is similar to the previous method. We only change the number of attributes since no suppression nor sampling is applied here.

Table 2.2: Experimental settings for DP with clustering.

setting	c	l	u	n	distribution
I. 25/L/U	500	0.25	0.75	25	100%: U(125,375)
II. 50/L/U	500	0.25	0.75	50	100%: U(125,375)
III. 25/S/U	2500	0.05	0.15	25	100%: U(125,375)
IV. 50/S/U	2500	0.05	0.15	50	100%: U(125,375)
V. 25/L/N	500	0	1.00	25	100%: N(250,100)
VI. 25/L/Un	500	0.25	0.75	25	75%: U(125,250), 25%: U(250,375)
VII. 200/L	100000	0.2	0.35	200	As used in [39]

In the evaluation, $\epsilon \in [0.5, 1, 2, 3, 4, 5]$ is the independent variable. We evaluate the performance with different confidence factors $c \in [0, 0.5, 0.7, 0.9]$. We use the upper bound for all the intervals as the input to the optimizer. Also, we carry out every experiment ten times and set a time limit of 1 minute for standard optimization.

PERFORMANCE ANALYSIS.

Figure 2.6 shows the performance of the differential privacy with clustering method. The approach with a low confidence factor shows a better objective ratio but lower feasibility. Moreover, all the approaches are robust with different distributions. When $c = 0$, the output data is $\{v_i + Lap(\Delta/\epsilon)\}$, which is also the average of the intervals when $c \neq 0$. When the value of c increases, the intervals become larger, and it is more probable that the accurate data is in the interval. As a result, the increasing upper bounds increase both the objective ratios and the feasibility. Also, the confidence factor can improve the feasibility at a small cost of the objective ratio when ϵ is small. For example, when $c = 0.7$,

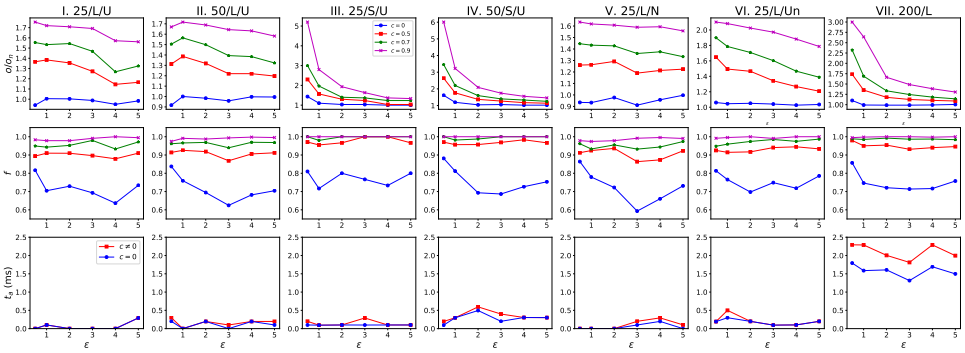


Figure 2.7: The performance of the comparison method (differential privacy without clustering). The x-axis is ϵ, and the y-axis is: the objective ratio o/o_n, the feasibility f, and the anonymization time t_a (ms).

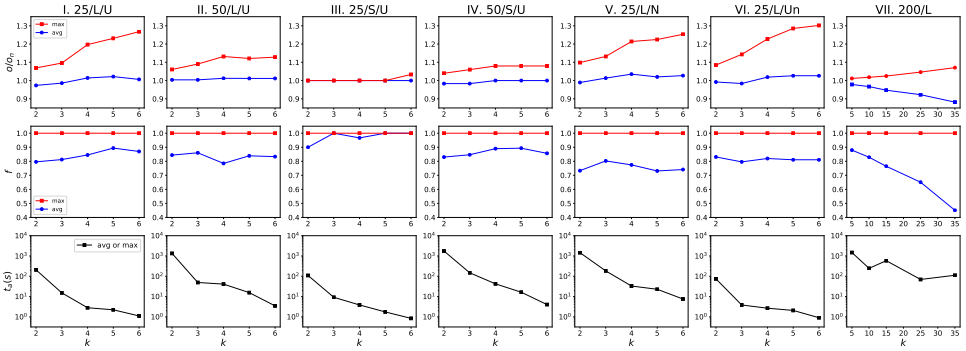


Figure 2.8: The performance of the comparison method (k-Optimize with standard optimizer). The x-axis is k, and the y-axis is: the objective ratio o/o_n, the feasibility f, and the anonymization time t_a (s).

the objective ratio is around 1.2, and the feasibility is around 0.9. Although the feasibility is not always equal to 1, we can mitigate it using a smaller capacity than the real capacity. Also, in practice, the trucks can remove some products to meet the constraints.

When ϵ increases, all the objective ratios are closer to 1, and all the feasibility increases. If ϵ keeps increasing, both the feasibility and the objective ratio can converge at 1. This shows a trade-off between the privacy concern and the utility for optimization. With a larger ϵ, less noise is added to the accurate data, so the algorithm has a weak privacy guarantee and good utility for the optimization work. Also, the anonymization can be finished within 3 ms.

2.6.5. COMPARISON RESULT

EXPERIMENTAL SETTINGS.

In this section, we compare the performance of our proposed methods to the differential privacy without clustering approach as used in the work of [31] (in Figure 2.7), and the work of [9] (in Figure 2.8), which applies two different k-anonymous algorithms (k-Optimize [11] and Flash [12]) to achieve privacy-preserving data publishing. The experi-

mental results show that k-Optimize shows the overall best performance [9], so we consider k-Optimize as the comparison method. Meanwhile, we set the minimum interval as **4** (e.g. $10 \rightarrow [8,14] \rightarrow [8,22]$...). A smaller minimum interval means a more optimized k-anonymous output, but the run-time becomes longer.

The instance setting is the same as the differential privacy with clustering method in Table 2.2. Considering the randomness from the input dataset, we carry out each comparison experiment **ten** times and use the average as results.

PERFORMANCE COMPARISON.

Figure 2.7 shows the result if only differential privacy is applied with confidence factors. It shows a similar result compared to the proposed differential privacy with clustering method. However, the clustering shows a better objective ratio. The objective ratio is around 1.2 when $c = 0.7$ (with clustering), but without clustering, the objective ratio is around 1.4 when $c = 0.7$, and even higher with an increasing number of items. Meanwhile, the feasibility is closer to 1 when clustering is applied. The result shows that the proposed clustering method can improve the original differentially private method in terms of objective ratio and feasibility. Compared to the differential privacy with k-anonymity method, our proposed method has better objective ratio (always between 1 and 1.2) and similar flexibility.

Figure 2.8 shows the performance of the k-Optimize method with the standard optimizer. We use both the average (avg) and the upper bound (max) of the intervals to show how well it works. The average shows a better objective ratio at the cost of the feasibility. The objective ratio using the upper bound of the k-Optimize output ranges from 1.1 to 1.3 for large items, and is very close to one for small items. Meanwhile, the feasibility of using the average values range from 0.8 to 0.9 for most settings, while it is very close to one for setting III and smaller than 0.8 for setting V. For all settings, a larger k always leads to an increase of the objective ratio since a larger k always means larger intervals in the output of the k-anonymous algorithm. The run-time for k-Optimize ranges from 10^0 to more than 10^3 seconds with 25 or 50 items.

The differential privacy with k-anonymity method and the k-Optimize method have shown very similar objective ratios and feasibility. Meanwhile, the differential privacy with k-anonymity method runs much faster than the k-Optimize, which means that we can expand the proposed method to 2-D or 4-D packing problems while k-Optimize can not. However, the proposed method is with suppression, while the k-Optimize considers all the input data. Because of the suppression, the number of rows of the input data is not the same for both methods, resulting in the differential privacy with k-anonymity method outperforms the k-Optimize. To better compare these methods, we compare the result of setting I for the proposed method in Figure 2.5 to the result of setting II for the proposed method in Figure 2.5. The proposed method is with fewer records in the input dataset, but it shows better feasibility and better objective ratio when $k \leq 4$. As a result, the proposed method can show a comparable result to the k-Optimize in terms of objective ratio and feasibility while it is much faster.

Compared to the differential privacy with clustering method, the k-Optimize method also shows a similar result. For example, when $\epsilon = 1$ and $c = 0.5$, the proposed method shows comparable objective ratios and better feasibility than the k-Optimize method

($k = 4$). With larger ϵ and smaller k, the proposed method also shows better objective ratios and feasibility than the k-Optimize method. With both the higher privacy guarantee or lower privacy guarantee, the proposed method can outperform or show comparable performance in terms of objective ratio and feasibility. Meanwhile, the proposed method is much faster.

2.7. CONCLUSIONS AND DISCUSSIONS

We propose two different privacy-preserving data publishing approaches using differential privacy to solve bin-packing problems under privacy-preserving. By calculating the probability of identifying the correct item, we prove that both proposed methods can provide better privacy guarantees than the previous work using k-anonymity. Using differential privacy, each item is supposed to be hidden among a group of items instead of only k items by using k-anonymity. Also, we carry out seven different experiments based on different data distributions and a different number of inputs. The results show that our proposed methods are much faster than the k-anonymous approach (from 10^3 s to less than 0.1 s) without any cost of objective ratio or feasibility. And the proposed methods are with better performance (lower objective ratio and higher or similar feasibility) than the approach only applying differential privacy. In conclusion, both proposed methods show advantages in privacy preservation and run-time over previous approaches that only apply k-anonymity or differential privacy while showing comparable objective ratio and feasibility. Meanwhile, both proposed methods can be used to solve 2-D or 4-D bin-packing problems, and we leave them as future works.

When we apply privacy-preserving methods, the better privacy guarantee always means the less useful output, so it is important to find the trade-off between these two aspects. In this paper, we use experiments to show the relationship between privacy guarantees (k and ϵ) and performance (o/o_n and f). With some performance cost (10%−20% o/o_n and f), the proposed methods can provide good privacy guarantees (such as $\epsilon = 1$). A better utility function or a better clustering method can help improve the performance of both proposed methods, and it remains as future works to find how much the utility function and the clustering can influence the performance factors.

REFERENCES

[1] T. Zhu, G. Li, W. Zhou, and S. Y. Philip. "Differentially private data publishing and analysis: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 29.8 (2017), pp. 1619–1638.

[2] Port of Rotterdam. *Port of Rotterdam throughput amounted to 469.4 million tonnes in 2019*. Accessed: 2021-02-18. 2020.

[3] M. Abdel-Basset, G. Manogaran, L. Abdel-Fatah, and S. Mirjalili. "An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems". In: *Personal and Ubiquitous Computing* 22.5 (2018), pp. 1117–1132.

[4] H. Feng, H. Ni, R. Zhao, and X. Zhu. "An enhanced grasshopper optimization algorithm to the Bin packing problem". In: *Journal of Control Science and Engineering* 2020 (2020).

[5] A. Van den Engel and E. Prummel. "Organised theft of commercial vehicles and their loads in the European Union". In: *European Parliament. Directorate General Internal Policies of the Union. Policy Department Structural and Cohesion Policies. Transport and Tourism, Brussels* (2007).

[6] M. Essig, M. Hülsmann, E.-M. Kern, and S. Klein-Schmeink. *Supply chain safety management*. Springer, 2013.

[7] *2020 Package Theft Statistics Report*. Accessed: 2021-09-21.

[8] ECMT. *Crime in Road Freight Transport*. OECD Publishing, 2002, p. 148.

[9] R. Hoogervorst, Y. Zhang, G. Tillem, Z. Erkin, and S. Verwer. "Solving bin-packing problems under privacy preservation: Possibilities and trade-offs". In: *Information Sciences* 500 (2019), pp. 203–216.

[10] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Technical Report, SRI International. 1998.

[11] R. J. Bayardo and R. Agrawal. "Data privacy through optimal k-anonymization". In: *21st International conference on data engineering (ICDE'05)*. IEEE. 2005, pp. 217–228.

[12] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. "Flash: efficient, stable and optimal k-anonymity". In: *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE. 2012, pp. 708–717.

[13] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. "l-diversity: Privacy beyond k-anonymity". In: *22nd International Conference on Data Engineering (ICDE'06)*. IEEE. 2006, pp. 24–24.

[14] N. Li, T. Li, and S. Venkatasubramanian. "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". In: *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*. IEEE Computer Society, 2007, pp. 106–115.

[15] K. B. Frikken and Y. Zhang. “Yet another privacy metric for publishing micro-data”. In: *Proceedings of the 2008 ACM Workshop on Privacy in the Electronic Society, WPES 2008, Alexandria, VA, USA, October 27, 2008*. ACM, 2008, pp. 117–122.

[16] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. “Differentially private combinatorial optimization”. In: *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2010, pp. 1106–1125.

[17] C. Zhang, M. Ahmad, and Y. Wang. “ADMM based privacy-preserving decentralized optimization”. In: *IEEE Transactions on Information Forensics and Security* 14.3 (2018), pp. 565–580.

[18] M. Zhang, Y. Chen, and W. Susilo. “PPO-CPQ: a privacy-preserving optimization of clinical pathway query for e-healthcare systems”. In: *IEEE Internet of Things Journal* 7.10 (2020), pp. 10660–10672.

[19] C. Dwork. “Differential Privacy”. In: *Automata, Languages and Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35908-1.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[21] F. McSherry and K. Talwar. “Mechanism Design via Differential Privacy.” In: *FOCS*. Vol. 7. 2007, pp. 94–103.

[22] N. Li, W. Qardaji, and D. Su. “On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy”. In: *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. 2012, pp. 32–33.

[23] E. C. man Jr, M. Garey, and D. Johnson. “Approximation algorithms for bin packing: A survey”. In: *Approximation algorithms for NP-hard problems* (1996), pp. 46–93.

[24] S. Martello. “Knapsack problems: algorithms and computer implementations”. In: *Wiley-Interscience series in discrete mathematics and optimiza tion* (1990).

[25] R. Agrawal and R. Srikant. “Privacy-preserving data mining”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 439–450.

[26] H. Lee and Y. D. Chung. “Differentially private release of medical microdata: an efficient and practical approach for preserving informative attribute values”. In: *BMC Medical Informatics and Decision Making* 20.1 (2020), pp. 1–15.

[27] X. Liu, Q. Xie, and L. Wang. “Personalized extended (α, k)-anonymity model for privacy-preserving data publishing”. In: *Concurrency and Computation: Practice and Experience* 29.6 (2017), e3886.

[28] L. Sweeney. “Datafly: A system for providing anonymity in medical data”. In: *Database Security XI*. Springer, 1998, pp. 356–381.

[29] K. LeFevre, D. J. DeWitt, R. Ramakrishnan, et al. “Mondrian multidimensional k-anonymity.” In: *ICDE*. Vol. 6. 2006, p. 25.

[30] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, et al. “A globally optimal k-anonymity method for the de-identification of health data”. In: *Journal of the American Medical Informatics Association* 16.5 (2009), pp. 670–682.

[31] N. Holohan, S. Antonatos, S. Braghin, and P. Mac Aonghusa. “(k, ϵ)-Anonymity: k-Anonymity with ϵ-Differential Privacy”. In: *arXiv preprint arXiv:1710.01615* (2017).

[32] A. Robinson, F. Brown, N. Hall, A. Jackson, G. Kemp, and M. Leeke. “CASTLEGUARD: Anonymised Data Streams with Guaranteed Differential Privacy”. In: *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE. 2020, pp. 577–584.

[33] M. Li, L. Zhu, Z. Zhang, and R. Xu. “Achieving differential privacy of trajectory data publishing in participatory sensing”. In: *Information Sciences* 400 (2017), pp. 1–13.

[34] Z. Ma, T. Zhang, X. Liu, X. Li, and K. Ren. “Real-time privacy-preserving data release over vehicle trajectory”. In: *IEEE Transactions on Vehicular Technology* 68.8 (2019), pp. 8091–8102.

[35] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. “Incognito: Efficient full-domain k-anonymity”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 2005, pp. 49–60.

[36] L. Perron and V. Furnon. *OR-Tools*. Version 7.2. Google. July 19, 2019. URL: https://developers.google.com/optimization/.

[37] M. Delorme, M. Iori, and S. Martello. “BPPLIB: a library for bin packing and cutting stock problems”. In: *Optimization Letters* 12.2 (2018), pp. 235–250.

[38] E. Falkenauer. “A hybrid grouping genetic algorithm for bin packing”. In: *Journal of heuristics* 2.1 (1996), pp. 5–30.

[39] A. Scholl, R. Klein, and C. Jürgens. “Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem”. In: *Computers & Operations Research* 24.7 (1997), pp. 627–645.

[40] M. Delorme, M. Iori, and S. Martello. “Bin packing and cutting stock problems: Mathematical models and exact algorithms”. In: *European Journal of Operational Research* 255.1 (2016), pp. 1–20.

[41] C. Dwork, A. Roth, et al. “The algorithmic foundations of differential privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

3

LOCATION DATA PERTURBATION

With the fast development of e-commerce, there is a higher demand for timely delivery. Logistic companies want to send receivers a more accurate arrival prediction to improve customer satisfaction and lower customer retention costs. One approach is to share (near) real-time location data with recipients, but this also introduces privacy and security issues such as malicious tracking and theft. In this paper, we propose a privacy-preserving real-time location sharing system including (1) a differential privacy based location publishing method and (2) location sharing protocols for both centralized and decentralized platforms. Different from existing location perturbation solutions which only consider privacy in theory, our location publishing method is based on a real map and different privacy levels for recipients. Our analyses and proofs show that the proposed location publishing method provides better privacy protection than existing works under real maps against possible attacks. We also provide a detailed analysis of the choice of the privacy parameter and their impact on the suggested noisy location outputs. The experimental results demonstrate that our proposed method is feasible for both centralized and decentralized systems and can provide more precise arrival prediction than using time slots in current delivery systems.

This chapter is a copy of the paper titled "Trajectory Hiding and Sharing for Supply Chains with Differential Privacy" by Li, T., Xu, L., Erkin, Z., and Lagendijk, R. L. in *28th European Symposium on Research in Computer Security (ESORICS 2023)*, pp. 297-317, 2023.

3.1. INTRODUCTION

Today, e-commerce is playing an important role in people's daily lives. According to *Statista*, in 2020, more than two billion people made orders online, with over $4.2 trillion in transactions. In e-retail, customers care about when they can receive the products, which raises the demand for logistics. Logistic companies, such as *DHL*, *UPS*, aim to minimize the delivery time while keeping packages safe [1]. Meanwhile, logistic companies provide a time slot for delivery. Unfortunately, these time slots usually span multiple hours, which reduces customer satisfaction on many levels [2]. On some occasions, the delivery time is updated to a new date and time due to transportation problems, causing frustrations and discomfort from customers. The mismatch between the predicted and actual arrival time causes problems for both customers and companies. Customers need to wait longer for the package. For companies, every delay adds to the cost of customer retention rate, customer acquisition cost, and customer lifetime value [3].

One possible solution is to provide a more precise delivery prediction, e.g. by offering real-time location data to help calculate the exact delivery time. According to *Hublock*, real-time location sharing systems are important and needed in logistics to improve the transparency of logistics. As a result, companies can improve customer satisfaction and lower the cost of retaining or acquiring customers [4]. Besides, the system is useful for disputes and knowing the reason for delays, and unburdening the customer service department [4]. It is already possible to see the use of real-time tracking, e.g., *DHL* offers a live tracking service for selected shipments [5]. Unfortunately, sharing accurate locations introduces security and privacy issues. According to [6, 7], the accurate location of trucks can be used for malicious tracking and theft. Imagine that a customer buys a very cheap product, locates the truck carrying that product, and steals other valuable packages in the same truck, resulting in economic damage [6, 7].

Given that we want to improve customer satisfaction by providing a realistic time of arrival and, at the same time, preventing potential theft, it is necessary to provide technical solutions that achieve both goals. There are existing approaches using generalization [8], adding dummy data [9], applying suppression [10], or using differential privacy [11] for publishing data with anonymity or privacy concerns. The first three methods are not suitable for real-time location sharing since they require the background knowledge of attackers and the whole trajectory as input, which are not available in real-time tracking since the entire trajectory is unknown when the truck is moving, and the adversary can carry out different attacks (e.g. malicious tracking or theft) based on background knowledge, such as the road map of the city. In contrast, differential privacy [12, 13] adds noise to the actual data and provides privacy guarantees, which is a strong candidate. Although there are existing approaches to publish location data with differential privacy [11, 14–16], there is no work considering both real-time location publishing and continuous trajectory privacy on a real map.

When the adversary holds real road maps, it is challenging to hide the trajectory of a truck. Even though the noise is added to real trajectory points, the published trajectory points are possibly up and down to the actual route, which can be de-noised using a filter or analysis. Meanwhile, it is important to add proper noise considering the road density. It is sufficient to add slight noise to anonymize the road for a truck moving with high road density, such as in the city centre. However, with the same noise, the actual

trajectory is distinguishable if the truck moves in an area with low road density, such as the countryside.

In this paper, we consider a network of logistic companies sharing location data with their customers using a location sharing platform. For different privacy-preservation needs and settings, protocols for centralized and decentralized platforms are needed. On the one hand, large enterprises can build their own centralized solutions. On the other hand, decentralized solutions are needed for small and medium-sized enterprises (SMEs), which occupy more than 90% of business in Europe [17]. SMEs often share similar needs but lack the technical resources to build or digitize their own supply chains. A platform shared by SMEs is desired to achieve the same functionality [18]. Blockchain is a candidate for the decentralized solution since it is traceable, immutable and transparent [19].

For trajectory hiding and secure location sharing, we focus on cities for package delivery and omit motorways. Location data of the Truck is reported based on regular intervals using the location sharing platform. The Sender and Receiver of a package in the Truck can access that information, which is used for estimating the time of arrival or any other optimization purposes. Note that using only the location perturbation algorithm cannot guarantee that the location is shared in a privacy-preserving manner on the platform. In order to provide protection, only the owner of a package and the corresponding delivery company should know the location information. We achieve this goal with cryptographic tools. Our proposal is effective regardless of the structure of the platform, which can be centralized or distributed, e.g. utilizing blockchain technology.

In summary, our contributions are as follows:

- We present a privacy-preserving location sharing system for logistics, including a location perturbation algorithm together with location sharing protocols, for tracking packages in (near) real-time to provide more precise arrival prediction than time slots. To the best of our knowledge, this is the first paper that considers real road maps and attacks for location perturbation.

- To prevent potential theft, we use differential privacy and geo-indistinguishability with different privacy levels for corresponding receivers. Our concrete privacy analysis and proof indicate the proposed approach provides better trajectory privacy preservation under real road maps and possible attacks than existing works. The detailed experiments show how privacy parameters are selected and how the utility remains in terms of arrival prediction. Also, the run-time is in the order of nanoseconds, which is feasible for real-time data sharing.

- To protect customers' privacy and the commercial interest of logistic companies, our proposed protocols provide anonymity, unlinkability and auditability in centralized and decentralized settings. Our experiments and analysis indicate that the proposed platform is privacy-preserving and has less storage cost than previous works. For feasibility, an Ethereum platform can process ~ 450 trucks due to the underlying blockchain technology, which is sufficient for average-sized cities even though the use of blockchain is not optimized.

Remark 3.1. The selection of platforms (blockchain) is not our focus since companies

can build their own centralized or decentralized solutions according to their needs with our proposed protocols.

3.2. PRELIMINARIES

Differential Privacy. Differential privacy (DP) was raised by Dwork [12, 13] to protect individual privacy and better use the dataset. In Equation 3.1, for neighbouring datasets, the probability of whether the output belongs to O differs less than e^{ϵ} with a small error factor δ, which hides the existence of any individual.

Definition 3.1 ((ϵ,δ)-differential privacy)**.** *An algorithm $\mathcal{A}$ satisfies (ϵ,δ)-differential privacy iff for neighbouring datasets D, D' which only differ in one record, and with any range $O \subseteq range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O] + \delta\,. \tag{3.1}$$

The Gaussian mechanism is a widely used mechanism to achieve (ϵ,δ)-differential privacy [20], which adds noise as $\mathcal{N}(\mu,\sigma)$ with $\mu = 0$, $\sigma^2 = 2\ln(1.25/\delta)\cdot(\Delta_2)^2/(\epsilon^2)$. δ is the small error, such as 10^{-5}. Δ_2 is the l_2 sensitivity.

Geo-Indistinguishability. Based on the definition of differential privacy, Andrés et al. [14] define geo-indistinguishability to allow to provide location based services (LBS) considering privacy within a radius r. In general, a mechanism $\mathcal{A}$ satisfies ϵ-geo-indistinguishability iff for any radius $r > 0$, the user enjoys ϵr-privacy within r, and the privacy level is proportional to r.

Definition 3.2 (geo-indistinguishability)**.** *An algorithm $\mathcal{A}$ satisfies ϵ-geo-indistinguishability iff for any two different points x, x':*

$$d_{\mathcal{P}}(\mathcal{A}(x),\mathcal{A}(x')) \leq \epsilon \cdot d(x,x')\,. \tag{3.2}$$

$d(\cdot,\cdot)$ denotes the Euclidean distance. For two different points x, x' s.t. $d(x,x') \leq r$, the distance $d_{\mathcal{P}}(\mathcal{A}(x),\mathcal{A}(x'))$ of corresponding distributions should be at most l, and $\epsilon = l/r$. Andrés et al. [14] present the Planar Laplace Mechanism which satisfies ϵ-geo-indistinguishability. Assume u is the smallest distance unit, δ_θ is the precision of the machine for angle θ, and r_{max} is the range within which the mechanism satisfies ϵ-geo-indistinguishability. If $q = u/r_{max}\delta_\theta$, we have ϵ from:

$$\epsilon' + \frac{1}{u}\ln\frac{q+2e^{\epsilon' u}}{q-2e^{\epsilon' u}} \leq \epsilon\,, \tag{3.3}$$

where ϵ' is the privacy parameter for $C_{\epsilon'}^{-1}(p)$. The noise is added to angle θ and distance r in Cartesian coordinates. $C_{\epsilon'}(r)$ shows the probability of any random point between 0 and r. If p is uniformly selected from $[0,1)$, we can get $r = C_{\epsilon'}^{-1}(p) = -\frac{1}{\epsilon'}\left(W_{-1}\left(\frac{p-1}{e}+1\right)\right)$ where W_{-1} is the Lambert W function.

3.3. SECURITY REQUIREMENTS

Objectives. The objective is a secure and privacy-preserving location sharing system for a number of trucks. On the one hand, the published location should have privacy preservation and good utility for arrival prediction. On the other hand, location data should be

published using a privacy-preserving platform. The platform only shares the location with Sender and Receiver, while no other information is leaked. More precisely, other parties in the platform cannot access the location of certain packages or link that package to a sender or a truck.

Set-up and Assumptions. There are three roles in the platform: **Truck** collects GPS data and shares it on the platform every n minutes. n is based on the number of Trucks simultaneously in the platform (considering system capacity) and how sparse the trajectory should be (considering privacy preservation). Only Trucks can publish information on the platform. **Sender** and **Receiver** access data from the platform, and each (Sender$_i$, Receiver$_i$) pair shares the same package information for package i. It is assumed that different companies share the same platform to provide location-based services to customers. Each company has several trucks but does not know the information of others. Moreover, we assume the distance to the destination is correlated to the delivery time. Other variables may also influence the estimate, including the characteristics of the road network and the current traffic levels. These are not considered here.

Adversary Model. In package delivery, we assume Trucks always send the correct location data, which is automatically collected from sensors and shared on the platform. Malicious drivers who can turn off the sensors are not considered. Internal adversaries (Senders and Receivers) can only access information from the platform. They try to misuse the available shared data from the platform to carry out malicious actions such as theft or malicious tracking. External adversaries try to steal the location data from the platform without access. Meanwhile, we assume the adversary has background knowledge of the truck, such as the road map of the city. However, we do not consider a powerful adversary with additional capacities, including surveillance cameras or drones. Such adversaries are hard to protect against even if no location information is shared.

Attack Model. There are possible attacks on the location perturbation process and the sharing platform. For location perturbation, adversaries try to re-identify the actual location of trucks by de-noising the published location data (such as using filters). With the identified location, adversaries can find the truck and carry out theft or malicious tracking. For the sharing platform, (1) adversaries try to find the linkage between customers and packages for malicious commercial analysis, such as finding target customers for certain logistic companies. (2) Adversaries try to get information about other packages. If adversaries know the location of all packages, they can find the target truck with target packages.

3.4. RELATED WORK

Location Privacy with DP. We consider DP-based location perturbation to provide privacy guarantees while publishing trajectory data in real-time. Dwork et al. [21] introduce the idea of event-level DP for DP under continual observation, but it is not robust when events are coming continuously. The actual location can be obtained by averaging the published location if the user stays in a certain area for a long time. Kellaris et al. [22] proposed ω-event DP to protect the event sequence occurring within ω successive timestamps by applying Laplace noise and budget allocation method. Fang et al. [23] gave the idea of δ-neighbourhood instead of the standard one. δ is a threshold for the generalized location point to guarantee that it is close to the actual location. Also, Xiao et al. [15]

proposed δ-location set based differential privacy to account for the temporal correlations and protect the accurate location at every timestamp. The temporal correlation is modelled through a Markov chain, and they hide the actual location in the δ-location set in which location pairs are indistinguishable. However, a reliable transition matrix is difficult to be constructed in a real scenario [24]. Xiong et al. [16] applied differential privacy to cluster and select location points, but the whole trajectory is known before the perturbation. Andrés et al. [14] gave the definition of geo-indistinguishability to allow location based services (LBS) to provide a service considering the privacy of individuals within a radius r. Also, the planar Laplace mechanism is proposed, which satisfies ϵ-geo-indistinguishability.

Although many different works consider location privacy, there are no works showing whether they can protect a real trajectory in a real use case with a real map under a possible attack. For example, suppose the trajectory of a truck is published and the adversary hold the background knowledge (e.g. the city map). In that case, the adversary may infer the actual location of the truck if there is only one road which the truck can pass around the published location.

Decentralized Supply Chains. Among decentralized solutions, blockchain is potentially a disruptive technology for supply chains since it is traceable, immutable, and transparent [19], with which the participants can trace the transaction. Maouchi et al. [25] proposed DECOUPLES, a decentralized, unlinkable, and privacy-preserving traceability system for supply chains. In their design, the PASTA protocol is proposed based on the stealth address to anonymize the receiver of a transaction. Each product has a unique product ID (pID). The receiver uses pID to generate a pair of tracking keys and sends the public key to the sender. The sender uses the public key to calculate a one-time stealth address as the receiver address, so only the receiver who owns the private key can track the package. However, they only consider two parties, while three parties (Sender, Truck, Receiver) are more common in real supply chains. This results in unnecessary one-time stealth addresses and more storage costs in real use.

Sahai et al. [26] proposed a privacy-preserving supply chain traceability system based on a protocol using zero-knowledge proofs and cryptographic accumulators. The proposed system provides unlinkability and untraceability, but only two parties are considered. Sezer et al. [27] designed a traceable, auditable, and privacy-preserving framework for supply chains using smart contracts. However, package information is not encrypted, which leads to possible leakage.

3.5. LOCATION PERTURBATION

3.5.1. PRIVACY PARAMETER SELECTION

In geo-indistinguishability, the privacy parameter ϵ controls how much noise is added to the location data. If the same amount of noise is added all the time, it is not large enough when the truck is far away from the destination and not small enough when close to the receiver, which influences the utility. The correct amount of noise should be added depending on the location of the truck. In the city centre, there are many routes within a small radius r, and it is possible to hide the real route with less noise. However, when the truck is located far away from the city centre, there are fewer alternative routes

Algorithm 3: Location Perturbation

Input: Current location x, destination location f, previous angle $\theta_0 = 0$
Output: Sanitized version z of input x
1: Get ϵ using Equation 3.6.
2: Get ϵ' using Equation 3.3.
3: $\theta \leftarrow AngleSelection(\theta_0)$, then set $\theta_0 \leftarrow \theta$.
4: Uniformly select $p \in [0,1)$ and set $r \leftarrow C_{\epsilon'}^{-1}(p)$.
5: $z \leftarrow x + < r\cos(\theta), r\sin(\theta) >$.
6: **return** z.

(consider a rural area with fewer roads around). To hide the real route, the radius r needs to be increased to include additional routes. Notice that we apply the distance to the city centre as the second factor for privacy parameter selection in this paper. Other factors, such as city density or road density, can also be used. We exclude motorways between cities since it is practically not possible to hide the location of a truck when there is only one road available.

With geo-indistinguishability where $l = \epsilon \cdot r$ (l is the privacy level, ϵ is the privacy parameter, and r is the radius). We can formulate l as:

$$l(x, f_i) = \begin{cases} l_s, & \text{if } d(x, f_i) \text{ is large} \\ l_m, & \text{if } d(x, f_i) \text{ is medium} \\ l_l, & \text{if } d(x, f_i) \text{ is small} , \end{cases} \tag{3.4}$$

where $d(x, f_i)$ is the distance between the location of truck x and receiver f_i. Here f_i represents the i-th receiver. A smaller privacy level (stronger privacy guarantee) is applied when the truck is far from the city centre, and l is larger to provide more precise arrival predictions when the truck is close to the receiver. The function is only applied when the delivery is scheduled for the next user i. Otherwise, l is set as l_s.

Similarly, r is based on the distance $d_i(x, c)$ between the truck (x) and the city centre (c). r should be smaller when the distance is shorter, so we have:

$$r(x, c) = \begin{cases} r_s, & \text{if } d_i(x, c) \text{ is small} \\ r_m, & \text{if } d_i(x, c) \text{ is medium} \\ r_l, & \text{if } d_i(x, c) \text{ is large} \end{cases} \tag{3.5}$$

$$\epsilon_i(x, f_i, c) = l(x, f_i)/r(x, c). \tag{3.6}$$

Here, the values of different parameters are chosen based on use cases. Different distance d and different privacy parameters ϵ should be defined based on the scenario. The selection of parameters is further discussed in Section 3.8.

3.5.2. ANGLE SELECTION

In geo-indistinguishability, only the privacy of single location points is considered without real road maps, as shown in Figure 3.1. The adversary can infer the actual trajectory even if every location point is protected. With a median filter and real maps, the

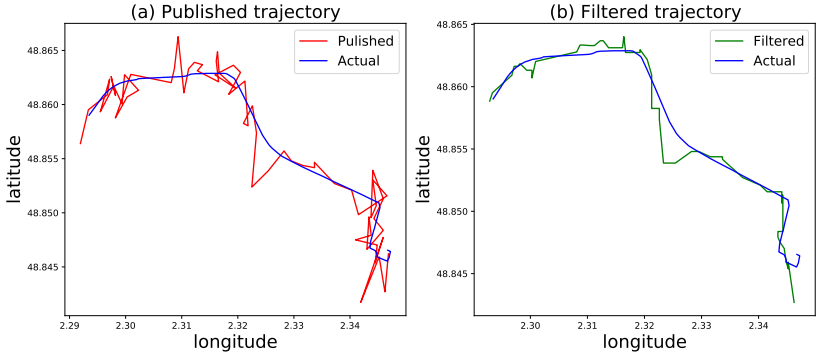


Figure 3.1: An example output by PL_ϵ, and the filtered output of the sanitized trajectory. The blue line shows the actual trajectory, the red line shows the published trajectory by PL_ϵ, and the green line shows the filtered trajectory.

Algorithm 4: AngleSelection

Input: Previous angle θ_0, privacy parameter ϵ_a
Output: Output perturbed angle θ
1: Calculate the new angle θ using Equation 3.7.
2: Round θ into the range $[0, 2\pi)$
3: **return** θ

adversary can achieve a trajectory close to the actual one (as shown in Figure 3.2). Although there are differences between the actual and published trajectories, adversaries can identify the correct road using a real map.

In this paper, we consider the connection between different location points by applying similar angles. Instead of uniformly selecting the new angle θ, we apply the Gaussian mechanism [20] to add noise to the previous θ_0, and

$$\theta = \theta_0 + \mathcal{N}(\mu = 0, \sigma = \sqrt{2\ln(1.25/\delta)} \cdot \Delta_2/\epsilon_a). \tag{3.7}$$

We use ϵ_a as the privacy budget for the angle selection mechanism to distinguish it from the ϵ for geo-indistinguishability. With Equation 3.7, we calculate the new angle θ and round it into the range $[0, 2\pi)$, as shown in Algorithm 4. The process mitigates the filtering attack by misleading the adversary to a wrong trajectory. We further analyze privacy protection in Sections 3.7 and 3.8.

Here the angle θ of round k_i is the input for round k_{i+1}. We need the composition theorem to calculate the privacy parameter ϵ_a with k rounds. In general, for k mechanisms M_i that all provide (ϵ, δ)-DP, the sequence of $M_i(x)$ provides $(k\epsilon_i, k\delta_i)$-DP [28]. By contrast, with the Gaussian noise, the scale is only $O(\sqrt{k})$.

Theorem 3.1. *For real-valued queries with sensitivity $\Delta > 0$, the mechanism that adds Gaussian noise with variance $(8k\ln(e + (\epsilon/\delta))\Delta_2^2/\epsilon^2)$ satisfies (ϵ, δ)-DP under k-fold adaptive composition for any $\epsilon > 0$ and $\delta \in (0, 1]$ [29].*

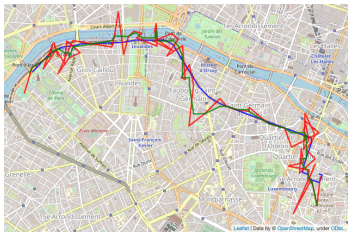


Figure 3.2: An example output by PL_ϵ on a real map. Blue: actual trajectory, red: published, green: filtered.

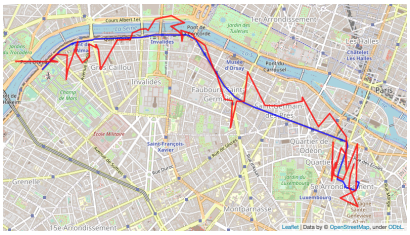


Figure 3.3: An example output after the Angle Selection is applied. Blue: actual trajectory. Red: published.

In Theorem 3.1, the variance for k-fold Gaussian mechanism is $(8\ln(e+(\epsilon/\delta))\cdot k\cdot \Delta_2^2/\epsilon^2)$ while for Gaussian mechanism is $(2\ln(1.25/\delta)\cdot\Delta_2^2/\epsilon^2)$. If we set the global privacy parameter as ϵ_0, the privacy parameter for each round is at the scale of $(\epsilon_0/\sqrt{k})$. In inverse, if each round is ϵ_a-DP, the angle selection algorithm provides $(\sqrt{k}\epsilon_a,\delta')$-DP where k is the number of rounds. The small error δ' is not further explored here, and we refer interested readers to [29].

Table 3.1: Notations used in protocols and algorithms.

NOTATION	DESCRIPTION
N_A, N_B	Nonce used by Alice and Bob
ID_A, ID_B	Identity of Alice and Bob
f	Key derivation function (KDF)
F_{AB}, F_{BA}	Key materials for KDF
K_{AB}	Long term key shared by Alice and Bob
K_{Aa}, K_{Ab}	Public key of Alice (or Bob)
k_{Aa}, k_{Ab}	Private key of Alice (or Bob)
p_{id}	Package ID
$TK_{p_{id}A}, TK_{p_{id}B}$	Tracking key of Alice and Bob for p_{id}
f_v	Flag for matching
H_s	Hash function
P	Stealth address
R	Transaction public key
P'	User-computed stealth address
E	AES-CBC encryption
E'	ECIES encryption

3.6. DECENTRALIZED LOCATION SHARING SYSTEM

In this section, we introduce the protocols for our decentralized location sharing system with notations in Table 3.1.

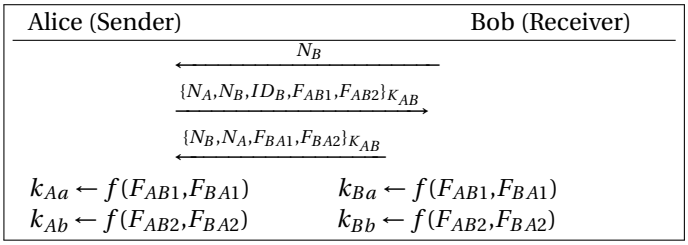
Initialization. With assumptions in Section 3.3, each Truck has an account (address). Companies register valid addresses at shared certificate owners (CO). The system

only accepts data from valid addresses and can track data accordingly.

Hiding Confidential Information. We encrypt the location data to provide confidentiality. A truck can transport several packages with the same location. Equation 3.6 implies only three possible location outputs. We apply AES-CBC to encrypt the logistic data. Then, we use the public key k_{Aa} for Elliptic Curve Integrated Encryption Scheme (ECIES) [30] to encrypt the symmetric keys. ECIES is based on Diffie-Hellman, with data and recipients' public keys as inputs.

Our Protocols. In PASTA [25] as in Section 3.4, for any specific package, Alice and Bob need two tracking keys to track the same data. Our design overcomes this shortcoming by sharing the same tracking key among them.

Firstly, Protocol 1 is used to establish a shared key between Alice and Bob. The protocol is based on the international standard ISO/IEC 11770-2-6 [31] where ID_i is the identity of i, N_A is a nonce (a number only used once), F_{AB} and F_{BA} are keying materials. Both Alice and Bob provide two key materials (for K_a and K_b). The session key is derived as $f(F_{AB}, F_{BA})$ where f is the key derivation function. K_{AB} is the long-term key shared by Alice and Bob.



Protocol 1. Key establishment mechanism. ID_i is the identity of i. N_A is a nonce. F_{AB}, F_{BA} are keying materials. f is the key derivation function. K_{AB} is the long-term key shared by Alice and Bob.

After the shared key is derived, Charlie (Truck) makes a request to both Sender and Receiver (Alice and Bob). For a (Truck, Sender, Receiver) pair, they share the same (p_{id}, TK_{pid}, K_b) and return the same keys ($K_{Ab} = K_{Bb} = K_b, TK_{pidA} = TK_{pidB} = TK_{pid}$) in Protocol 2. TK is the tracking key, p_{id} is package id shared between the sender and receiver, H_s is a hashing function.

Protocol 3 shows how the three-party stealth address works. Using the public shared keys TK and K_b, Truck can generate a random r and broadcast the (R, P) pair. For the Sender and Receiver, they can calculate the stealth address P' using private tracking key tk_{pid} and R and find the match on the platform.

By applying the proposed protocols, each location record is shared with both Receiver and Sender instead of storing two same records on the decentralized platform, so in theory, we can save half storage than the PASTA protocol as used in [25].

Charlie (Truck)	Alice (or Bob)
ID_A, ID_B, p_{id}	
	Request $TK_{p_{id}A}$ and $TK_{p_{id}B}$ →
	$TK_{p_{id}A} \leftarrow H_s(p_{id}k_{Aa})G$
	$K_{Ab} \leftarrow k_{Ab}G$
	← $TK_{p_{id}A}, K_{Ab}$
if $(TK_{p_{id}A} \stackrel{?}{=} TK_{p_{id}B}$	
and $K_{Ab} \stackrel{?}{=} K_{Bb})$:	
Protocol 3	

Protocol 2. Matching function to check whether Alice and Bob return the same keys. TK is the tracking key, p_{id} is package id, H_s is a hashing function.

Charlie (Truck)	Alice (or Bob)
$r \in_R Z_p, R \leftarrow rG$	
$P \leftarrow H_s(rTK_{p_{id}})G$	
$+K_b$	
	Broadcast (R,P) →
	Check each (R,P)
	$tk_{p_{id}} \leftarrow H_s(p_{id}k_{Aa})$
	$P' \leftarrow H_s(tk_{p_{id}}R)G$
	$+K_{Ab}$
	$P' \stackrel{?}{=} P$

Protocol 3. Three-party stealth address protocol. TK and K_b are public shared keys. P is the stealth address. r is the random nonce.

Extensions. The proposed protocols can be used for centralized platforms or blockchain-based platforms. For centralized platforms, with Protocols 1, 2, the encrypted location information can be shared. For blockchain platforms, the certificate owner (CO) uses *register transaction* to control the validity of trucks. Trucks use *publish transaction* to publish real-time location data. With the contract, we can validate and trace the source of a transaction. For a *register transaction*, we verify the sender is a valid CO and the value is valid. For a *publish transaction*, the contract checks the validity of the sender and the data.

3.7. ANALYSIS

3.7.1. SECURITY AND PRIVACY ANALYSIS

Location Perturbation. Section 3.2 includes different trajectory publishing mechanisms, but most only consider differential privacy in theory, and all the approaches do not consider a real map. There are existing works [32] showing that a differentially private mechanism still suffers from attacks in real use cases. In this paper, with assumptions in Section 3.3, we consider privacy under real maps, showing that our proposed approach provides better privacy protection.

Figures 3.2–3.5 show example outputs from PL_ϵ and the proposed method for two different trajectories. In Figures 3.2 and 3.4, the filtered trajectory is close to the actual, and it is predictable on which road the truck is moving. Although the trajectory is in a large city, Paris, it is hard to hide from the actual. The basic idea of our proposed angle selection approach is to mislead the adversary to a wrong trajectory that is close to the actual one but not the same one. If the noise trends in the same direction (e.g. south) as the actual trajectory, the adversary can identify the wrong road. In Figure 3.5, when the published trajectory is south to the real one, it is more probable for the adversary to infer the wrong road. Moreover, we conclude Lemma 3.2, indicating that the proposed angle selection mechanism achieves stronger privacy guarantees than randomly selecting angles.

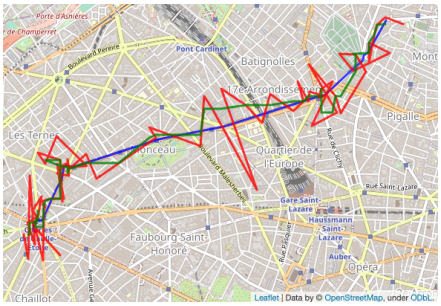


Figure 3.4: Another example output by PL_ϵ. Blue: actual, red: published, green: filtered.

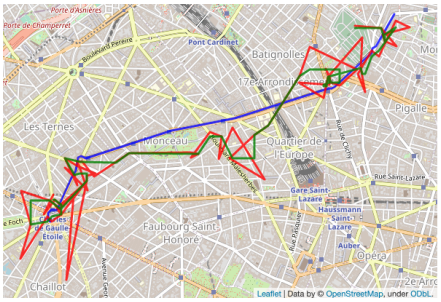


Figure 3.5: Another example output with angle selection. Blue: actual, red: published, green: filtered.

Lemma 3.2. *The angle selection mechanism can provide stronger privacy protection than randomized angle selection, considering trajectory hiding in real maps under attacks (such as median filters).*

Proof. In this proof, we first introduce our metrics to evaluate privacy protection levels under median filter attacks. The analysis is generally applicable, and we use Figure 3.6 to visualize how the metrics work under attacks. Then, we show that a larger distance perturbation results in better location privacy protection. Finally, we prove that the proposed angle selection mechanism provides stronger privacy guarantees than existing works.

Figure 3.6 shows an example trajectory with three location points $La_0(x_0, y_0)$, $La_1(x_1, y_1)$, $La_2(x_2, y_2)$. Similarly, outputs with angle selection are Ls_0, Ls_1, Ls_2, and with random angles are Lr_0, Lr_1, Lr_2. Ma_{01}, Ms_{01}, Mr_{01} are the midpoint for the first two location points (such as La_0 and La_1). We can compare privacy protection levels between different approaches using two metrics when a median filter is applied: (1) **distance difference**. Compare the distance between the published location points (midpoints), for example, the distance between Ma_{01} and Ms_{01} to the distance between Ma_{01} and Mr_{01}. (2) **length of average vector differences**. Compare the distance of average vector difference from $Ma_{01}Ma_{12}$ to $Ms_{01}Ms_{12}$ and from $Ma_{01}Ma_{12}$ to $Mr_{01}Mr_{12}$. The average vector difference shows the distance difference among the published trajectories (as the dotted lines) since the lines can intersect in the middle. Here the average vector difference from $Ma_{01}Ma_{12}$ to $Ms_{01}Ms_{12}$ is $\frac{1}{2}(\overrightarrow{Ma_{01}Ms_{01}} + \overrightarrow{Ma_{12}Ms_{12}})$.

Assume an adversary $\mathcal{A}$ knows the perturbation is generated from the Laplace distribution. With the published location (Lp_i) and the distance $d_i \geq 0$, the probability that $\mathcal{A}$ can identify the original location (La_i) can be calculated. The probability that the guessed distance $d_x \geq \Delta$ is within Δ to the actual distance d_i is:

$$\begin{aligned}\frac{\int_{d_i-\Delta}^{d_i+\Delta} p(d_x)dd_x}{\int_0^{\infty} p(d_x)dd_x} &= \frac{\int_{d_i-\Delta}^{d_i+\Delta} \frac{1}{b}\exp(-\frac{d_x}{b})dd_x}{\int_0^{\infty} \frac{1}{b}\exp(-\frac{d_x}{b})dd_x} = \exp\left(-\frac{d_i-\Delta}{b}\right) - \exp(-\frac{d_i+\Delta}{b}) \\ &= \exp\left(-\frac{d_i}{b}\right)\cdot\left(\exp\left(\frac{\Delta}{b}\right) - \exp\left(-\frac{\Delta}{b}\right)\right).\end{aligned} \tag{3.8}$$

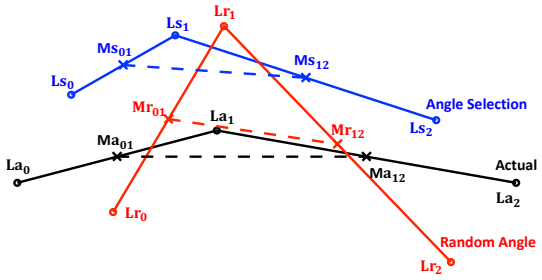


Figure 3.6: An example trajectory with three location points.

where b is the scale. With the same error Δ, Equation 3.8 shows that a smaller distance d_i means a higher probability of guessing a more accurate approximate perturbation distance. The adversary can draw a circle with a radius close to the distance to infer the actual location with a real road map. A circle with a larger radius can cover more roads, so it is harder to locate the actual location and privacy is better protected.

In Algorithm 3, the perturbation for $La_0(x_0, y_0)$ is $(r\cos(\theta), r\sin(\theta))$, $\theta \in [0, 2\pi)$. The output is $L_0(x_0 + r_0\cos(\theta_0), y_0 + r_0\sin(\theta_0))$. Similarly, we have La_1 and L_1. If the midpoint for L_0L_1 is M_{01}, we have Ma_{01} and $M_{01}(x_{01}, y_{01})$. For distance difference, we have the distance d_{01} between Ma_{01} and M_{01} that

$$\begin{aligned} 4 \cdot d_{01}^2 &= (x_{01} - (x_0 + x_1))^2 + (y_{01} - (y_0 + y_1))^2 \\ &= (r_0\cos(\theta_0) + r_1\cos(\theta_1))^2 + (r_0\sin(\theta_0) + r_1\sin(\theta_1))^2 \\ &= r_0^2 + r_1^2 + 2r_0r_1\cos(\theta_0 - \theta_1). \end{aligned} \tag{3.9}$$

With the same amount of noise (the same r_0, r_1), we can maximize d_{01} when $\theta_0 = \theta_1$. By the angle selection mechanism, θ_0 has a higher probability of being closer to θ_1 than randomly selected, resulting in a larger d_{01} and stronger privacy guarantee. Similarly, we have the average vector difference $v_d(x_d, y_d)$ as:

$$2v_d = \overrightarrow{Ma_{01}M_{01}} + \overrightarrow{Ma_{12}M_{12}} \tag{3.10}$$

We have $4x_d = (r_0\cos(\theta_0) + 2r_1\cos(\theta_1) + r_2\cos(\theta_2))$ and $4y_d$ similarly. We can calculate the length of the average vector difference $|v_d|$ from $16|v_d|^2$ as:

$$\begin{aligned} &(r_0\cos(\theta_0) + 2r_1\cos(\theta_1) + r_2\cos(\theta_2))^2 + (r_0\sin(\theta_0) + 2r_1\sin(\theta_1) + r_2\sin(\theta_2))^2 \\ &= r_0^2 + 4r_1^2 + r_2^2 + 4r_0r_1\cos(\theta_0 - \theta_1) + 4r_1r_2\cos(\theta_1 - \theta_2) + 2r_0r_2\cos(\theta_0 - \theta_2). \end{aligned} \tag{3.11}$$

To maximize $|v_d|$, we have $\theta_0 = \theta_1 = \theta_2$. The angle selection mechanism lets every output θ_i similar to the previous angle θ_{i-1}, which results in a larger $|v_d|$.

The proposed system achieves larger distance and vector differences under filter attacks with the same amount of added noise. With Equation 3.8, the angle selection mechanism provides stronger privacy guarantees than random selection. □

The angle selection mechanism satisfies $(\sqrt{k}\epsilon_a, \delta')$-DP for k rounds, which means that angles are hidden among the range of $[0, 2\pi)$ with privacy budget $\sqrt{k}\epsilon_a$. In Section 3.2, we assume location points are published every n minutes. Considering half-day

Table 3.2: Computational and storage analysis. N_T, N_L, N_P: number of trucks, encrypted location data, destined product information. k_{SE}: key size (bits) for AES-CBC. $e, a, (p, r)$: size (bits) of encrypted data, address, stealth address.

Protocol	Operation	Truck	Receiver/ Sender	On-Chain Storage
Protocol 1	Key Derivation	-	$\mathcal{O}(N_P)$	-
Protocol 2	Tracking Key Derivation	-	$\mathcal{O}(N_P)$	-
Protocol 3	Compute Stealth Address	$\mathcal{O}(N_P)$	-	-
Confidential Data Sharing	Decryption Encryption	$\mathcal{O}(eN_P)$ -	- $\mathcal{O}(eN_P)$	- -
Smart Contract Operation	Register	$\mathcal{O}(N_T)$	-	aN_T
	Publish	$\mathcal{O}(N_L + N_P)$	-	$N_L e + N_P(k_{SE} + (p, r))$

delivery with six hours and $n = 5$, there are $k = 72$ rounds and $\sqrt{k} \approx 8.5$. With a total desired privacy budget ϵ_{all}, $\epsilon_a = \epsilon_{all}/\sqrt{k}$ is for each round. The noise is added to angles, so the output is probably beyond the range $[-\pi, \pi)$. This can lead to a random output angle with a small $\epsilon_a (< 1)$. To achieve higher utility, we select a larger ϵ_a to output an angle with higher probability in the range of $[-\pi, \pi)$. With a larger ϵ_a, the adversary may infer that the perturbed angle is related to the previous angle. Differential privacy (DP) has the strong assumption that the adversary knows all other records in the dataset, but the adversary never knows any output angle in our scenario. It is secure to select a larger ϵ_a, such as $\epsilon_a = 5$. Section 3.8 shows how we select ϵ_a. From the definition of DP, the angle selection results in a larger privacy parameter than selecting uniformly, but Lemma 3.2 illustrates it can provide stronger privacy guarantees against real adversaries with possible attacks. It is not sufficient to only consider privacy guarantees based on the definition of DP. Instead, a stronger adversary with background knowledge should be considered since this is non-negligible in real cases. Other DP-based works [15, 22, 23] also consider location privacy similarly with analysis only in theory or lines instead of a real map.

The privacy parameter selection function provides different privacy guarantees based on distances under real maps. In a real use case, receivers only need a more precise location when the truck is close. If the receiver is far away from the city, there can be privacy leakage, and it is easy to identify the road of the truck (since there might be only one route within a small radius). Meanwhile, the delivery prediction error can be larger than hours when the package is far from the receiver, but when the truck is within k km, the error should be minimized (to minutes). With the privacy parameter selection function, we can better protect the real location of trucks and provide a more precise arrival time prediction when the truck is away or close to the receiver.

Location Sharing System. Our encryption algorithm relies on the security of AES-CBC and ECIES encryption functions. For Protocol 1, the international standard ISO/IEC 11770 [31] guarantees Alice and Bob can securely exchange key materials. The key derivation function PBKDF2 [33] guarantees only the holders of key materials can generate the key k. The security and privacy of protocol 2 are based on the assumption that SHA-3 is a cryptographically secure hash function. If a probabilistic polynomial time (PPT)

adversary $\mathcal{A}$ obtains the private tracking key $tk_{p_{idA}} = H_s(p_{id}k_{Aa})$, $\mathcal{A}$ can not derive k_{Aa} or identity of the owner since the hash function is one-way. The security and privacy of protocol 3 rely on ECDLP [34]: given two points $P, Q \in E(\mathbb{F}_p)$ where $Q \in < P >$, finding a k such that $Q = kP$ is computationally infeasible. Meanwhile, protocol 3 holds the property of anonymity and unlinkability (with proof in Appendix 3.10.1).

Lemma 3.3. *(Anonymity and unlinkability) A PPT adversary $\mathcal{A}$ can not derive the receiver of a stealth address or distinguish the receiver of two different stealth addresses in Protocol 3.*

Remark 3.2. If an adversary aims to access 100 trajectories from multiple days and trucks, he needs to send or receive 100 packages. Also, the 100 trajectories will not follow the same routes since the receivers are not the same.

3.7.2. PERFORMANCE ANALYSIS

We analyze our protocols with a blockchain-based platform to show the feasibility and performance since blockchain is a potentially disruptive technology for supply chains [19]. We summarize the computation complexity and on-chain storage in Table 3.2, showing that the computation complexity is linear with the number of trucks or packages. Meanwhile, the proposed encryption method has a lower storage cost than DECOUPLES [25] (with proof in Section 3.10.2).

The protocols can also be used for centralized platforms where the complexity is only determined by Protocols 1, 2, which is less, but a trusted and reliable centre is needed to avoid possible hardware failure or information leakage [35].

3.8. EXPERIMENTAL EVALUATION

3.8.1. LOCATION PERTURBATION

This subsection includes the selection of privacy parameters (ϵ, ϵ_a), and evaluation (run time and distance difference). We use Python for implementation, with Mac OS 11, 2 GHz Quad-Core Intel Core i5 CPU, 16 GB RAM.

Dataset. The GPS trajectory dataset (collected by GeoLife) [36] is used for evaluations. Trajectories are collected by different GPS loggers and GPS phones from 182 users, including 17,621 trajectories covering 1,292,951 kilometres. We have evaluated our algorithms using different trajectories and we use each trajectory to simulate one stop of the truck based on the map of Beijing.

Distance Metric. We use the Haversine formula as the error function to calculate the distance difference between two location points. If φ and λ are latitudes and longitudes, and r is the radius of the Earth, we have $d((\varphi_1, \lambda_1), (\varphi_2, \lambda_2))$ as:

$$d = 2r \arcsin \sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}. \tag{3.12}$$

$\boldsymbol{\epsilon}$ *(for ϵ-geo-indistinguishability)* is selected by Equation 3.6. We can define which distance is large, medium, or small based on city sizes. For example, inner, central, and outer rings in cities define the distance to the centre. For the first run of the algorithm,

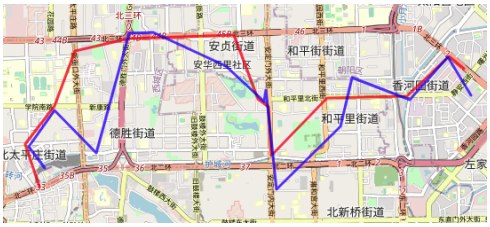


Figure 3.7: Example output with $\epsilon = 0.01$.

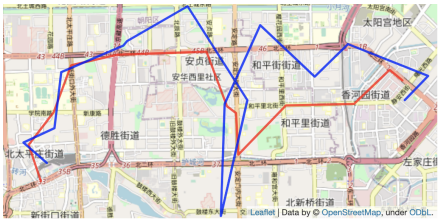


Figure 3.8: Example output with $\epsilon = 0.005$.

Figure 3.9: Example output with $\epsilon = 0.0025$.

Figure 3.10: Example output with $\epsilon = 0.001$.

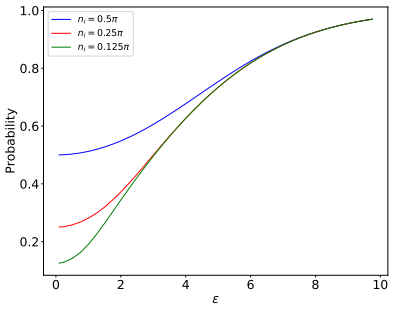


Figure 3.11: Relation between the output probability and ϵ when the angle noise n_i is 0.125π, 0.25π and 0.5π.

we need to scale the privacy level l with the output results using different r. Table 3.3 shows the relation between r and the real distance difference, so we can calculate l by multiplying the average distance and ϵ. When $l \approx 3.2$, the distance difference is approximately the same as r. We set $l_m = 3$, $l_s = 1$ (to better preserve location privacy by lowering ϵ) and $l_l = 5$. Considering the density of roads in a city, we can set r to contain at least n (such as 5) different roads with different distances between the truck and city centre. Here we set $r_m = 1000(m)$, $r_s = 400$, $r_l = 2000$ using the real road map of Beijing. Figures 3.7–3.10 support that the proposed parameters work well in the real map with different (l, r) pairs. For example, with $l = 3, r = 400$, we have $\epsilon = 0.0075$, whose output is similar to Figure 3.7 and 3.8. If $l = 1, r = 1000$, we have $\epsilon = 0.001$ as shown in Figure 3.10 with much larger noise. After defining parameters l and r for the first time, the value of ϵ can be calculated in real uses.

ϵ_a *(for angle selection)* can be selected based on the probability of outputting an angle ranging in $(\theta_0 - n_i, \theta_0 + n_i)$ where $n_i \in [-\pi, \pi)$ is the output noise of the Gaussian mechanism. We can draw the output probability in Figure 3.11 using:

$$p(n_i) = \int_{-n_i}^{n_i} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} dx \Big/ \int_{-\pi}^{\pi} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} dx \tag{3.13}$$

where $\sigma = \sqrt{2\ln(1.25/\delta)} \cdot \Delta_2/\epsilon_a$ and Δ_2 is the l_2-sensitivity. With different n_i, the same ϵ_a results in a similar probability when $\epsilon_a > 4$, so we can choose the desired ϵ_a (such as 5) and control the probability (such as 0.7).

Distance Difference. Based on Equation 3.12, we evaluate the average distance (the dis-

tance between the actual location and the published one) and the average distance error (the error for the calculation of the distance between the current location to the destination) in Table 3.3. All experiments are performed 100 times based on the dataset while the average is used. A smaller ϵ has a larger error, meaning that the distance or the error is smaller when the truck is closer to the receiver (small l) and the city centre (small r).

Run Time. The run time is around 10^{-8} seconds ($< 1\ \mu s$). The proposed algorithm can be applied to smart devices to sanitize location data in real-time.

Utility Analysis. Figures 3.7–3.10 and Table 3.3 illustrates the relation between ϵ and the distance error. With a small $\epsilon = 0.0005$, the average distance error is around 4.18 km. If the truck speed is at 50 km/h, considering the distance error is the straight-line distance (without considering road maps), the actual arrival time prediction error is around 5 to 10 minutes. However, for the adversary, Table 3.3 shows that the distance difference is 6.43 km. Even if they know that the truck is within 6.43 km of the published location, they need to check the circle area with a radius of $r_0 = 6.43\ km$ to find the truck. With our proposed angle selection mechanism, the adversary needs to check the roads in $\pi r_0^2 = 129.9\ km^2$ to find the truck, which is infeasible in practice. In Figure 3.10 with $\epsilon = 0.001$, the published location is several streets away from the original location. The adversary cannot locate the truck even if they hold the road map. Similarly, with a large $\epsilon = 0.01$, the difference or error is only around 200 meters, which infers that the prediction error is within one minute. Figure 3.7 shows that the published trajectory is close to the actual, but the angle selection method can mislead the adversary to the south of the real trajectory. Moreover, a large ϵ is only set when the truck is close to the receiver in the city centre.

3.8.2. LOCATION SHARING SYSTEM

We implement and evaluate our protocols with Ethereum to test the feasibility of our protocols. In real cases, enterprises can choose their own solutions based on the proposed protocols. We use *Rust* for implementation and *JavaScript VM* to deploy the smart contract. ChaChaRng is the pseudo-random number generator. SHA-3 is the hash function. Curve25519 is the elliptic curve. AES-CBC is with a 128-bit key. All tests are with Win 10 Pro, 32GB RAM, and Intel Core i7-10700.

We evaluate the run time for our protocols (where S/R is Sender/Receiver): (1) key derivation (S/R: 0.506 s), (2) generate TK_{pid} (S/R: 0.438 ms), (3) generate stealth address P (Truck: 0.850 ms), and (4) generate user-computed stealth address P' (S/R: 0.440 ms). The key derivation limits the performance. The off-chain encryption includes (i) AES-CBC to encrypt the data and (ii) ECIES to encrypt the symmetric keys. The run time for ECIES (0.295 ms) is much longer than AES (172 ns) with $N_L = 20, N_P = 100$, which limits the performance. With 30 items and stealth addresses (512-bit), the average gas cost for our encryption method is 2.398×10^6, which is less than DECOUPLES [25] (2.864×10^6).

The scalability relies on the proof of work consensus model. For every second, Ethereum can process around 15 transactions [37], so our platform can publish location data from 15 trucks. Assume the location data is sent every five minutes. The platform can support $15 \times 60 \times 5 = 450$ trucks, which is practical for SMEs.

Table 3.3: Average distance and average distance error in meters with different ϵ.

$\epsilon = l/r$	Avg. distance	Avg. error	$\epsilon = l/r$	Avg. distance	Avg. error
0.0001	31661.92	27017.11	0.0005	6435.09	4180.74
0.001	3231.63	1963.17	0.003	1076.24	619.72
0.005	656.07	371.39	0.006	532.53	309.11
0.007	453.70	265.82	0.008	401.11	232.70
0.01	319.22	184.90	0.05	63.55	37.16

3.9. CONCLUSIONS

We propose a real-time privacy-preserving location sharing system considering real maps and possible filtering attacks. We improve the state-of-the-art in two folds. Firstly, our proposed location publishing mechanism is feasible in real applications. Based on our exclusive security and privacy argumentation and proof, the proposed angle selection algorithm can better protect the privacy of trajectories than existing works. The experiments show the location publishing method is fast and practical for real-time data processing, which only needs nanoseconds. Secondly, our proposed location sharing protocols can protect privacy-sensitive data using cryptographic constructions under centralized and decentralized settings. Our security analysis proves that the system is privacy-preserving. With Ethereum, our proposal has lower storage costs compared to the previous work [25]. It is feasible and can handle ~450 trucks, a reasonable amount for an average city. Companies can build their own solutions using our protocols to improve.

3.10. APPENDIX

3.10.1. PROOF FOR LEMMA 3.3

Lemma 3.4. *(Anonymity and unlinkability) A PPT adversary $\mathcal{A}$ can not derive the receiver of a stealth address or distinguish the receiver of two different stealth addresses in Protocol 3.*

Proof. Assume that a PPT adversary $\mathcal{A}$ holds a stealth address (P,R) and p_{id} and a list of tuples $(TK_{i,p_{id}}, K_{b_i})$, $\mathcal{A}$ needs to compute $P' = H_s(rTK_{i,p_{id}})G + K_{b_i})$ such that $P' = P$. To find such a P', $\mathcal{A}$ need to compute $P - K_{b_i} = H_s(rTK_{i,p_{id}})G$. Because of the one-wayness of ECDLP, it is computationally infeasible to compute the $H_s(rTK_{i,p_{id}})$. And since $\mathcal{A}$ does not know the secret value r, he can not contrust $P' = H_s(rTK_{i,p_{id}})G + K_{b_i})$ himself. Therefore, it is infeasible for $\mathcal{A}$ to derive the receiver of (P,R).

Similarly, assume that $\mathcal{A}$ gets two stealth addresses (P_1,R_1) and (P_2,R_2), $\mathcal{A}$ needs to distinguish the following two scenarios: (1) two stealth addresses belong to the same receiver, and (2) two stealth addresses belong to two different receivers. For scenario (1), $\mathcal{A}$ computes $P_1 - P_2$ as:

$$\begin{aligned} P_1 - P_2 &= H_s(rTK_{p_{id1}})G + K_b - (H_s(rTK_{p_{id2}}) + K_b) \\ &= (H_s(rTK_{p_{id1}}) - H_s(rTK_{p_{id2}}))G \\ &= xG \quad \text{for some unknown x.} \end{aligned} \tag{3.14}$$

Since the adversary $\mathcal{A}$ does not hold p_{id1}, p_{id2} and r, $(H_s(rTK_{p_{id1}}) - H_s(rTK_{p_{id2}}))$ is a secret value x for him. For scenario (2), $\mathcal{A}$ computes $P_1 - P_2$ as:

$$\begin{aligned} P_1 - P_2 &= H_s(r_1TK_{p_{id1}})G + K_{b_1} - (H_s(r_2TK_{p_{id2}}) + K_{b_2}) \\ &= (H_s(r_1TK_{p_{id1}}) - H_s(r_2TK_{p_{id2}}) + K_{b_1} - K_{b_2})G \\ &= yG \quad \text{for any unknown y.} \end{aligned} \tag{3.15}$$

The adversary $\mathcal{A}$ does not hold p_{id1}, p_{id2}, r_1, r_2, so $(H_s(r_1TK_{p_{id1}}) - H_s(r_2TK_{p_{id2}}) + K_{b_1} - K_{b_2})G$ is a secret for $\mathcal{A}$.

In both scenarios, the adversary $\mathcal{A}$ can not derive the secret value. Given two different stealth addresses, it is computationally infeasible for $\mathcal{A}$ to distinguish. □

3.10.2. PROOF OF LOWER STORAGE COST

Lemma 3.5. *The proposed encryption method has lower storage costs than DECOUPLES [25].*

Proof. The space cost for only using ECIES is $S_{ECIES} = N_P(e + (p,r))$. To compare the space cost of the encryption algorithm S and S_{ECIES}, we compute $S - S_{ECIES}$ as follows:

$$\begin{aligned} S - S_{ECIES} &= N_L e + N_P(k_{SE} + (p,r)) - N_P(e + (p,r)) \\ &= (N_L - N_P)e + N_P(k_{SE} - e) \end{aligned} \tag{3.16}$$

Since many products share the same location, we have $N_L < N_P < 0$. If $e > k_{SE}$, we get $S - S_{ECIES} < 0$ (the size of the encrypted data is larger than the size of the symmetric key). Our encryption method requires less storage than ECIES. □

REFERENCES

[1] A. E. Branch. *Global supply chain management and international logistics*. Routledge, 2008.

[2] N. A. H. Agatz, A. M. Campbell, M. Fleischmann, and M. W. P. Savelsbergh. "Time Slot Management in Attended Home Delivery". In: *Transp. Sci.* 45.3 (2011), pp. 435–449.

[3] Auditshipment. *The True Cost of Package Delivery Delays*. https://www.auditshipment.com/blog/the-true-cost-of-package-delivery-delays/. Accessed: 2021-11-7. 2021.

[4] M. Grmiling. *How Real Time Tracking Can Improve Logistics*. https://www.hublock.io/how-real-time-tracking-can-improve-logistics/. Accessed: 2021-11-17. 2021.

[5] DHL. *Parcel delivery in real time*. https://www.dhl.de/en/privatkunden/pakete-empfangen/sendungen-verfolgen/live-tracking.html. Accessed: 2022-01-07. 2021.

[6] A. Van den Engel and E. Prummel. "Organised theft of commercial vehicles and their loads in the European Union". In: *European Parliament. Directorate General Internal Policies of the Union. Policy Department Structural and Cohesion Policies. Transport and Tourism, Brussels* (2007).

[7] E. U. SAVONA. "Organised Property Crime in the EU". In: *European Parliament. Directorate General for Internal Policies. Policy Department for Citizens' Rights and Constitutional Affairs*. (2020).

[8] N. Harnsamut, J. Natwichai, and S. Riyana. "Privacy Preservation for Trajectory Data Publishing by Look-Up Table Generalization". In: *ADC 2018*. Vol. 10837. LNCS. Springer, 2018, pp. 15–27.

[9] S. Hayashida, D. Amagata, T. Hara, and X. Xie. "Dummy Generation Based on User-Movement Estimation for Location Privacy Protection". In: *IEEE Access* 6 (2018), pp. 22958–22969.

[10] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos. "Local Suppression and Splitting Techniques for Privacy Preserving Publication of Trajectories". In: *IEEE Trans. Knowl. Data Eng.* 29.7 (2017), pp. 1466–1479.

[11] H. Wang and Z. Xu. "CTS-DP: Publishing correlated time-series data via differential privacy". In: *Knowl. Based Syst.* 122 (2017), pp. 167–179.

[12] C. Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35908-1.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[14] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. "Geo-indistinguishability: differential privacy for location-based systems". In: *ACM CCS 2013*. ACM, 2013, pp. 901–914.

[15] Y. Xiao and L. Xiong. “Protecting Locations with Differential Privacy under Temporal Correlations”. In: *ACM CCS 2015*. ACM, 2015, pp. 1298–1309.

[16] P. Xiong, T. Zhu, L. Pan, W. Niu, and G. Li. “Privacy Preserving in Location Data Release: A Differential Privacy Approach”. In: *PRICAI 2014*. Vol. 8862. LNCS. Springer, 2014, pp. 183–195.

[17] S. Brunswicker and V. Van de Vrande. “Exploring open innovation in small and medium-sized enterprises”. In: *New frontiers in open innovation* 1 (2014), pp. 135–156.

[18] L. Wong, L. Leong, J. Hew, G. W. Tan, and K. Ooi. “Time to seize the digital evolution: Adoption of blockchain in operations and supply chain management among Malaysian SMEs”. In: *Int. J. Inf. Manag.* 52 (2020), p. 101997.

[19] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen. “Blockchain technology and its relationships to sustainable supply chain management”. In: *Int. J. Prod. Res.* 57.7 (2019), pp. 2117–2135.

[20] C. Dwork, A. Roth, et al. “The algorithmic foundations of differential privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[21] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. “Differential privacy under continual observation”. In: *STOC 2010*. ACM, 2010, pp. 715–724.

[22] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. “Differentially Private Event Sequences over Infinite Streams”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1155–1166.

[23] C. Fang and E. Chang. “Differential privacy with δ-neighbourhood for spatial and dynamic datasets”. In: *ASIA CCS 2014*. ACM, 2014, pp. 159–170.

[24] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar. “Location Privacy-preserving Mechanisms in Location-based Services: A Comprehensive Survey”. In: *ACM Comput. Surv.* 54.1 (2022), 4:1–4:36.

[25] M. E. Maouchi, O. Ersoy, and Z. Erkin. “DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain”. In: *SAC 2019*. ACM, 2019, pp. 364–373.

[26] S. Sahai, N. Singh, and P. Dayama. “Enabling Privacy and Traceability in Supply Chains using Blockchain and Zero Knowledge Proofs”. In: *IEEE International Conference on Blockchain, Blockchain 2020, Rhodes, Greece, November 2-6, 2020*. IEEE, 2020, pp. 134–143.

[27] B. B. Sezer, S. Topal, and U. Nuriyev. “An Auditability, Transparent, and Privacy-Preserving for Supply Chain Traceability Based on Blockchain”. In: *CoRR* abs/2103.10519 (2021). arXiv: 2103.10519.

[28] F. McSherry. “Privacy integrated queries: an extensible platform for privacy-preserving data analysis”. In: *SIGMOD 2009*. ACM, 2009, pp. 19–30.

[29] P. Kairouz, S. Oh, and P. Viswanath. “The Composition Theorem for Differential Privacy”. In: *ICML 2015*. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1376–1385.

[30] V. Shoup. “A Proposal for an ISO Standard for Public Key Encryption”. In: *IACR Cryptol. ePrint Arch.* (2001), p. 112.

[31] ISO. *ISO/IEC 11770-2:2008, Information technology – Security techniques – Key Management – Part 2: Mechanisms using Symmetric Techniques*. International standard. International Organization for Standardization, 2009.

[32] B. Hitaj, G. Ateniese, and F. Pérez-Cruz. “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning”. In: *ACM CCS 2017*. ACM, 2017, pp. 603–618.

[33] K. M. Moriarty, B. Kaliski, and A. Rusch. “PKCS #5: Password-Based Cryptography Specification Version 2.1”. In: *RFC* 8018 (2017), pp. 1–40.

[34] D. Hankerson, A. J. Menezes, and S. Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[35] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K. R. Choo. “Sidechain technologies in blockchain networks: An examination and state-of-the-art review”. In: *J. Netw. Comput. Appl.* 149 (2020).

[36] Y. Zheng, L. Zhang, X. Xie, and W. Ma. “Mining interesting locations and travel sequences from GPS trajectories”. In: *WWW 2009*. ACM, 2009, pp. 791–800.

[37] I. A. Seres, D. A. Nagy, C. Buckland, and P. Burcsi. *MixEth: efficient, trustless coin mixing service for Ethereum*. Cryptology ePrint Archive, Report 2019/341. 2019.

4

LOCATION DATA SHARING WITH PRIVACY PRESERVATION

Privacy preservation is challenging for digitized supply chains in light of regulatory constraints such as GDPR. While GPS data is crucial for trajectory tracking in supply chains, it can inadvertently expose the carrier's location, violating GDPR guidelines. This paper presents PrivTrack, which leverages a differentially private trajectory perturbation algorithm and cryptographic protocols for privacy-preserving trajectory tracking for blockchain-based supply chains with constrained IoT devices. First, we utilize an efficient differentially private algorithm for trajectory hiding against a more practical adversary model with real road maps and possible filter attacks. We optimize it to be adaptable for deployment on constrained IoT devices. Our detailed evaluation demonstrates its feasibility in real-world scenarios and privacy protection against possible de-noising attacks. Second, we propose a platform for privacy-preserving trajectory sharing with cryptographic protocols. Our IoT-based solution is anti-spoofing without human interaction. Our proof-of-concept blockchain-based solution also provides data validation while the smart contract validates the authenticity in a decentralized manner and allows access control. Furthermore, our evaluation shows that our solution is cost-effective, as the execution time of our protocols is ~200 ms with an autonomy of almost one month, which is feasible for real-world use cases. Our flexible and scalable solution empowers companies to customize their solutions by selecting and varying privacy parameters, storage options, and IoT devices while maintaining all properties.

This chapter is a copy of the paper titled "PrivTrack: Privacy-Preserving Trajectory Tracking for Supply Chains" by Li, T., Kromes, R., Erkin, Z., and Lagendijk, R. L., which is under review from *IEEE Transactions on Dependable and Secure Computing.*

4.1. INTRODUCTION

In January 2017, the electronic Contract for the International Carriage of Goods by Road (eCMR) [1] was officially launched and used in European Union logistics. The purpose of the CMR Convention is to standardize the conditions of the contract for cross-border carriages of goods. eCMR is the electronic version of freight documents, which replaces the use of old paper CMR to increase efficiency and reduce paperwork in logistics. The occurrence of eCMR increases the need for digitized supply chains. However, most digitized solutions rely primarily on centralized systems, such as cloud or local back-end services, which suffer from potential data manipulation and unauthorized access to the data [2]. As a consequence, financial institutions suffered an average of $5.72 million in losses in 2021 [3].

Centralized systems cannot provide entirely trustworthy data computing, traceability or immutable data storage. In contrast, blockchain is a distributed and immutable ledger with traceability, non-repudiation and transparency. Such properties are important to protect against counterfeit products and improve transparency. Also, the blockchain-based platform can be shared among small and medium-sized enterprises (SMEs) to save costs and enhance cooperation. Data validation and privacy are of great concern in a blockchain-based supply chain [4]. Note that logistic data is commercially sensitive, and enterprises do not want their data known to other participants, so a privacy-preserving platform is desired [5]. Though blockchain provides traceability and immutability, data validation is important since a dishonest party can easily add fake data to the blockchain and fool the users. Among different blockchain-based supply chain platform solutions [4, 6–9], security and privacy usually rely on advanced cryptographic protocols. However, such designs lack data validation and a general system architecture with different participant roles.

Besides a reliable digitized supply chain, one other crucial factor for eCMR is the (geo)location of trucks. During transportation, location data is needed for delivery inspection and cross-border checks. Meanwhile, logistics companies also need to share location data with their customers to increase customer satisfaction and provide more precise arrival prediction [7, 10]. In practice, companies, such as *DHL*, share the actual location of trucks with their customers [11]. Customers want to acquire the location data to estimate a more precise arrival time and understand the possible reason for delays. Though the location of trucks is highly demanded by both logistics and customers, the location of truck drivers is considered sensitive information according to GDPR, implying that the actual location should not be shared [12]. Sharing location data can raise the risk of drivers since they will be the target when a target package is contained in the truck.

Considering location privacy, there are several existing solutions [13–18] where differential privacy is a promising technique for efficient real-time trajectory hiding without knowing any background knowledge for the attacker. However, they only consider privacy protection in theory based on the differential privacy guarantees. Even with their trajectory perturbation algorithms, it is unclear whether the approaches can protect against a more practical adversary model with background knowledge and possible de-noising attacks. In contrast, Li et al. [7] consider privacy protection under possible filter attacks, and the adversary holds background knowledge of the trajectory, such as

the real road map. The approach includes the privacy parameter selection for a proper choice of ϵ for geo-indistinguishability, together with the angle and distance perturbation schemes to add noise to a location point in the polar coordinates.

Most of the previously mentioned solutions are designed and tested on mobile phones, which have powerful computation abilities. However, mobile phones suffer from possible GPS spoofing [19]. If the holder of the mobile phone is malicious, the GPS information can be altered before being sent, which can not be validated for the use of supply chains. Using constrained IoT devices can be a promising solution for performing location privacy protection operations without human interaction. The device is considered to be isolated during the transportation of goods, which avoids possible physical intrusion.

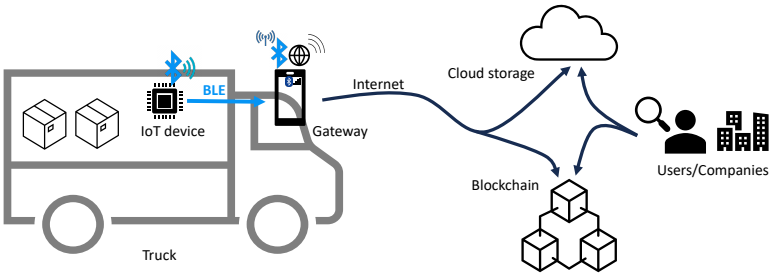


Figure 4.1: Our designed trajectory tracking solution for supply chains.

The application of constrained IoT devices faces different challenges when preserving location privacy. Firstly, constrained IoT devices are limited in terms of computational power, memory size, and battery life. Since location sharing is a dynamic process, it implies that the constrained IoT device is equipped with a battery and uses wireless communication for data transmission. With existing works, it is unclear whether it is feasible to achieve location privacy in constrained IoT devices and how the performance is impacted. Also, considering the transportation can last over several days in a supply chain use case, a study exploring the device's battery life is also required. Furthermore, it is challenging to ensure data authenticity in the constrained IoT context, especially when the IoT device needs to be authenticated by a blockchain [20, 21]. The creation of a valid blockchain transaction or the digital signature generation for a payload performed locally in the IoT device is a resource-intensive computation [22, 23]. Finally, for constrained IoT devices, the security requirements may differ from those of mobile phones or computers. The differences involve the random number generation for the trajectory perturbation algorithm and the elliptic curve-based cryptographic operations for blockchain interactions such as digital signature and Diffie-Hellman key exchange (DHKE).

In addition, blockchain can be utilized to achieve data integrity and authenticity, which are validated in a decentralized manner through smart contracts. The sensing IoT devices can also be authenticated in a decentralized way to remove the need for a trusted third party that might misbehave during authentication. Moreover, the smart contract allows the deployment of sophisticated access controls to verify which client

has access to the data and add new data to be stored in the blockchain.

Our objective is to develop a privacy-preserving trajectory tracking platform for supply chains against adversaries, ensuring secure data sharing, data validation and location privacy. Additionally, the platform should be feasible for real-world use with the proposed protocols on constrained IoT devices.

We assume that the platform is shared among a group of companies (SMEs) for location based services to their customers. Companies have their own trucks but do not know the information of other companies. There are six roles in our design (as shown in Figure 4.1), with the following actions. The logistics company initializes a transportation session. The IoT device collects location data, and it sends the hash to the blockchain and the encrypted data to the cloud storage through the gateway. The private keys of IoT devices are not shared with the logistics companies and are maintained by the manufacturer of the device, or the secret key is hardware secure, meaning it cannot be accessed. We assume the truck driver does not maliciously turn off the gateway. Such malicious behaviour can not be prevented even without privacy concerns, but it should be detected based on the design. When a package is delivered, the truck driver collects proof of the delivery, such as the signature of the recipient. The blockchain is used to ensure data authenticity and validation in a decentralized manner. The cloud is used for data storage. The logistics company and users can access and validate the data via smart contracts. Users are the customers whose packages are transported in the truck. For the location perturbation algorithm, we assume the distance to the destination is correlated to the delivery time. In this paper, we do not consider other variables that may influence the estimate, such as the characteristics of the road network and the current traffic levels.

In our adversary model, the IoT device is honest. It collects and sends the logistics data to the gateway. The gateway is malicious, and it tries to tamper with the data. Note we assume that truck drivers cannot turn off the gateway, and they register the delivery of packages upon delivery proof. Otherwise, these malicious actions can be detected. The users are malicious and try to misuse the logistics information to locate the truck driver during transportation by de-noising the published location data. The malicious users hold background knowledge of the truck, such as the road map of the city, but they are not powerful enough to access surveillance cameras or drones. Such over-powerful adversaries can not be protected against even if no location information is shared. Also, malicious users try to infer the information of other recipients in the same truck and the location of other trucks without having any package inside so that they can locate the target truck carrying the target packages. The logistics company is semi-honest when initializing transportation (Generating and sharing correct keys). For other phases, we assume it is malicious, trying to deceive customers by tampering or adding invalid logistics data. The blockchain platform is operated in a decentralized manner, and the cloud storage is operated by trusted parties.

In this paper, we propose **PrivTrack**, with contributions in two categories: (1) a trajectory perturbation algorithm and (2) a privacy-preserving trajectory sharing platform with constrained IoT devices.

Trajectory Perturbation Algorithm. We utilize differential privacy and geo-indistinguishability to mislead the adversary to a wrong trajectory, which protects against pos-

sible filter attacks under real road maps. Our proposal relies on [7], which proposes a trajectory hiding mechanism consisting of three parts: privacy parameter selection, angle perturbation and distance perturbation. However, we improve the privacy parameter selection part of that work to be more adaptable for deployment in constrained IoT devices with less storage, battery, and time costs. Our evaluation shows the feasibility of deploying the algorithm on a constrained IoT device. Our further experiments demonstrate that the perturbed location data is robust against filter attacks and remains useful for eCMR and package tracking.

Trajectory Sharing Platform. For privacy-preserving data sharing, we design a blockchain-based solution with cryptographic protocols to ensure data validity, confidentiality, integrity and availability with detailed security and privacy analysis. Our proposed PoC platform supports data validation with blockchain and constrained IoT devices. Data and IoT devices are authenticated in a decentralized way to remove the need for a trusted third party. Moreover, our smart contract allows access control to verify which client has access to the blockchain and the cloud storage. Furthermore, we have an implementation (that will be published on GitHub upon paper acceptance) that enables location privacy and blockchain interaction for constrained IoT devices through gateway communication. We also provide an implementation to call hardware secure computation of cryptographic operations to guarantee different security requirements. The location privacy operation entails a reasonable cost in terms of energy consumption and execution time. We present the main performances for different processing states of our proposed protocol when higher security requirements in secret key operations are applied.

We claim that our proposed solution is flexible and scalable for companies to customize their solutions with varying privacy parameters, storage options, and IoT devices while retaining all properties.

The remainder of our paper is organized as follows. In Section 4.2, we introduce necessary backgrounds, including the related works on location privacy, blockchain-based supply chains and constrained IoT devices. In Section 4.3, we explain our trajectory perturbation algorithm based on differential privacy and geo-indistinguishability. In Section 4.4, we show the design of our protocols and system architecture based on cryptographic protocols. In Sections 4.5 and 4.6, we demonstrate implementation details and evaluation results for both the trajectory perturbation algorithm and the proposed data-sharing platform based on constrained IoT devices. In Section 4.7, we give the security and privacy analysis for the trajectory perturbation algorithm and the system design. In Section 4.8, we conclude our work.

4.2. BACKGROUND AND RELATED WORK

In this section, we introduce the necessary background and related works, including abundant existing solutions for location privacy, privacy-preserving blockchain-based supply chain, and the application of IoT in blockchain use cases. Based on the literature, it is unclear whether location privacy and data validation can be achieved via constrained IoT devices in real-world uses. Moreover, a general framework is missing for privacy-preserving data sharing and tracking within a supply chain when IoT devices

are considered.

4.2.1. LOCATION PRIVACY WITH DIFFERENTIAL PRIVACY

For location privacy, differential privacy [24, 25] is widely used and aims to hide the existence of any record in a dataset by applying a small noise to the query.

Definition 4.1 ((ϵ,δ)-differential privacy)**.** *For neighbouring datasets D, D' which only differ in one record, an algorithm $\mathscr{A}$ satisfies (ϵ,δ)- differential privacy iff with any range $O \subseteq$ range($\mathscr{A}$):*

$$\Pr[\mathscr{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathscr{A}(D') \in O] + \delta\,. \tag{4.1}$$

For example, the Gaussian mechanism achieves (ϵ,δ)-differential privacy [26] by adding Gaussian noise to the query. The noise is $\mathcal{N}(\mu,\sigma)$ with $\mu = 0$, $\sigma^2 = 2\ln(1.25/\delta)\cdot(\Delta_2)^2/(\epsilon^2)$. δ is a small error, Δ_2 is the l_2 sensitivity. In this paper, the Gaussian mechanism is applied for the location perturbation.

There is plentiful research with differential privacy for location privacy, such as $\omega-$event DP [27], $\delta-$neighbourhood [14], geo-indistinguishability [13], $\delta-$location set [18], etc., but existing solutions only consider privacy protection in theory based on the definition of differential privacy. Besides, Li *et al.* [7] achieve trajectory hiding with real road maps by applying angle selection and privacy parameter selection to the idea of geo-indistinguishability. The authors [7] design a location-sharing platform for logistics based on Ethereum, but this is expensive for the gas cost and inefficient for practical use that the blockchain only supports ~60 trucks. For applications, most approaches only consider mobile phones [28, 29], which may suffer from spoofing attacks. There also exist efficient algorithms that claim they can be used on IoT devices, but they lack evaluation on IoT platforms, such as [30, 31].

Among various approaches, geo-indistinguishability, as introduced in [13], has low computation cost and is tested with real road maps against possible filter attacks in [7]. Let set $\mathcal{X}$ contain all possible user location points, and set $\mathcal{Z}$ include all possible returned location points. $d(\cdot,\cdot)$ denotes the Euclidean distance.

Definition 4.2 (ϵ-geo-indistinguishability)**.** *An algorithm $\mathscr{A}$ satisfies ϵ-geo-indistinguishability iff for $x, x' \in \mathcal{X}$, $Z \subseteq \mathcal{Z}$:*

$$\mathscr{A}(x)(Z) \leq e^{\epsilon d(x,x')}\mathscr{A}(x')(Z)\,. \tag{4.2}$$

The Planar Laplace Mechanism satisfies ϵ-geo-indistinguishability, as shown in Figure 6 in [13] where the noise is added in polar coordinates. The perturbed angle θ is uniformly drawn from $[0,2\pi)$ and the perturbed distance r is

$$C_{\epsilon'}^{-1}(p) = -\frac{1}{\epsilon'}\left(W_{-1}\left(\frac{p-1}{e}\right)+1\right) \tag{4.3}$$

with p uniformly drawn from $[0,1)$. W_{-1} is the Lambert W function. The mechanism satisfies ϵ-geo-indistinguishability where

$$\epsilon \geq \epsilon' + \frac{1}{u}\ln\frac{q+2e^{\epsilon' u}}{q-2e^{\epsilon' u}}\,. \tag{4.4}$$

Here u and δ_θ are precision parameters for r and θ. r_{max} is the range for geo-indistinguishability, and $q = u/r_{max}\delta_\theta$. For more details, we refer to [13].

4.2.2. PRIVACY PRESERVATION IN BLOCKCHAIN-BASED SUPPLY CHAINS

Blockchain is deemed a disruptive technology for supply chains due to its traceability and immutability. However, privacy and data validation are not usually well addressed. For example, modum.io [6] is presented to monitor the temperature and humidity during transportation in a pharmaceutical supply chain. The data is collected by IoT devices and then shared with the public via blockchain, but privacy is not considered in the design, and the data source is not validated.

With privacy concerns, Wu *et al.* [9] propose a framework with multiple private ledgers and one public ledger. The private ledger is used for customers of a specific shipment to share custody events. The public ledger is for global tracking, including the geolocation of trucks. The consensus is based on proof of work, and the load is increased due to private and public chains. Privacy is considered in the private ledger, but the privacy of geolocation is not addressed. Furthermore, data validation is done by crowd-sourcing from the participants in the private chain to ensure that they share the same information. This still can not deal with malicious users who add fake data or events to the chain. Besides, the work of [4] shows a design for freight declaration and tracking. The logistics data is validated via different kinds of claims. Actual geolocation is shared with customers for tracking, which leaks information. Maouchi *et al.* [8] design a privacy-preserving blockchain-based traceability system for supply chains, which achieves receiver anonymity and unlinkability. The protocol is based on the stealth address, where transactions are always broadcast. In practice, this can restrict its application in permissioned blockchains. Also, the practical factors are not included, such as data collection, location privacy, and actor roles.

In industry, the Tradelens [32] project was initiated by IBM to deploy a sophisticated blockchain-based product shipment tracking system under access control. However, the project was halted due to a lack of interest from potential commercial partners in investing [5]. Another example is Walmart's Food Trust project, which is an IBM-based blockchain service with the goal of tracing the origins of 20 different alimentary products [33]. Both projects are strictly based on Hyperledger consortium blockchains and are not scalable for different types of blockchain platforms. Moreover, a solution with IoT and gateway communication is not provided and implemented. Data validation is not considered, and the location privacy of deliverers is not addressed in their proposals.

4.2.3. BLOCKCHAIN WITH CONSTRAINED IOT DEVICES

For integrating IoT with blockchain technology, the authors of [23, 34] insist that IoT devices should establish direct, authenticated communication with the blockchain network. Therefore, blockchain transactions are created and signed locally on the IoT device, and the validity of the transaction is proved by the smart contracts. Since the transactions are signed on the IoT devices, data authenticity is reinforced. Furthermore, if a gateway is involved in the communication to forward the transaction to the blockchain, the gateway can not alter the transaction since the digital signature of the transaction needs to be validated by the blockchain. With the digital signatures of the transactions, in addition to data authenticity, more traceable and trustworthy data processing is guaranteed using the smart contract. The approach cannot be considered a generic solution since the IoT device can only communicate with a single blockchain due to the differ-

Algorithm 5: Location Perturbation Algorithm $\mathscr{A}_{LP}$

Input: Current location point x, previous angle $\theta_0 = 0$, city map $\mathcal{M}$, ϵ_a for angle selection.
Output: Sanitized version z of input x

procedure PRIVACY ($\mathcal{M}$)
  Define privacy parameters for different districts in $\mathcal{M}$.
  Get ϵ' using Equation 4.5 with x, $\mathcal{M}$.
end procedure
procedure ANGLE (θ_0, ϵ_a)
  Get the new angle θ using Equation 4.6 with θ_0, ϵ_a.
  Round θ into the range $[0, 2\pi)$, and then set $\theta_0 \leftarrow \theta$.
end procedure
procedure DISTANCE (ϵ', θ)
  $r \leftarrow C_{\epsilon'}^{-1}(p)$ with p uniformly selected from $[0, 1)$.
  $z \leftarrow x + < r\cos(\theta), r\sin(\theta) >$.
end procedure
return z.

ent requirements for transaction creation, such as transaction structure, type of elliptic curve in the digital signature, and the hash algorithm used. Arnaudo *et al.* [22] proposed a generic protocol for allowing IoT communication with different types of blockchains. The protocol enables a generic payload to be digitally signed using the ECDSA scheme and secp256k1 curve. Afterwards, the signed payload is sent to the gateway, which incorporates it into the desired blockchain transaction. In case the gateway alters the payload, the smart contract denies its execution, and the payload is not registered.

4.3. TRAJECTORY PERTURBATION ALGORITHM

In this section, we introduce our location perturbation algorithm $\mathscr{A}_{LP}$, as shown in Algorithm 5, using privacy parameter selection, geo-indistinguishability and angle selection based on [7]. Our approach optimizes the deployment of $\mathscr{A}_{LP}$ on constrained IoT devices. In $\mathscr{A}_{LP}$, we choose the privacy parameter based on the city road map, and we add noise in terms of angle and distance. The perturbed angle for a new round is similar to the previous round so that the output series of location points is more probably perturbed to the same direction (such as to the south) of the real trajectory, where the Gaussian mechanism is applied to generate the angle. The perturbed distance is generated similarly to the Planar Laplace Mechanism.

4.3.1. PRIVACY PARAMETER SELECTION

According to geo-indistinguishability, the amount of noise for perturbation is controlled by the privacy parameter ϵ. In practice, if the same noise is added all the time, it might be too large for an area with high road density and be too small with low road density. For example, in the city centre with potential high road density, a circle within a small radius r may cover a number of different streets. As a result, a small noise is sufficient to

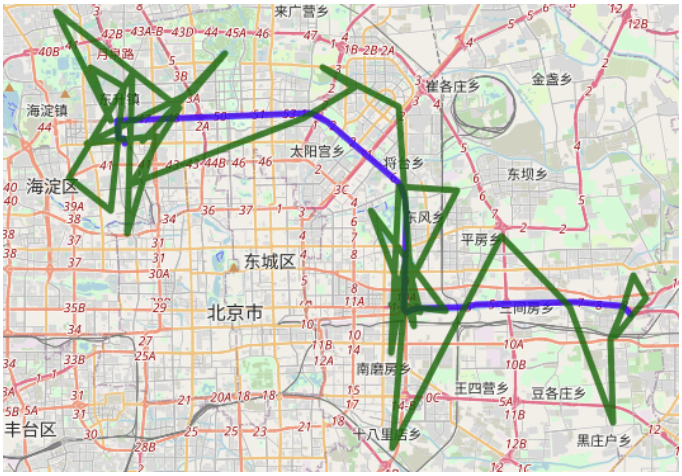


Figure 4.2: An example output by PL_ϵ on a real map with $\epsilon = 0.001$. Blue: actual trajectory, green: published with PL_ϵ.

preserve privacy. However, when a truck is moving in a rural area, there may be only a few roads nearby, and the actual road can be identified with the same perturbation. As a result, the privacy parameter should be determined based on the relative location of the truck regarding the city road map $\mathcal{M}$.

In large cities, the city is usually divided into several areas. For example, Paris has 20 administrative districts. Similarly, Shanghai has four ring expressways according to the distance to the city centre. Using such urban planning information, we can determine different privacy parameters ϵ for different locations of the truck x based on the city road map $\mathcal{M}$, as shown in Equation 4.5.

$$\epsilon(x, \mathcal{M}) = \begin{cases} \epsilon_s, & \text{if } x \text{ is in suburb area } \mathcal{M}_s \\ \epsilon_i, & \text{if } x \text{ is in District } \mathcal{M}_i \\ \epsilon_c, & \text{if } x \text{ is in central area } \mathcal{M}_c \end{cases} \tag{4.5}$$

From the equation, ϵ_s is the smallest (with the largest noise) since it is in a rural area with fewer roads nearby. Similarly, ϵ_c is the largest, considering the high road density in the city centre. For different district $\mathcal{M}_i$, the value of epsilon should be selected based on different scenarios. The PRIVACY procedure in Algorithm 5 shows the angle selection algorithm, and we further address the selection of privacy parameters in Section 4.5.

Remark 4.1. The selection of privacy parameters can also be dependent on other factors, such as the distance between the truck and the recipients, but this can lead to higher computation and communication costs for IoT devices to send multiple transactions to different receivers. We further analyze the complexity in Section 4.7.

4.3.2. ANGLE PERTURBATION

Geo-indistinguishability considers location privacy for single location points, while trajectory privacy is not well addressed under real road maps. In geo-indistinguishability,

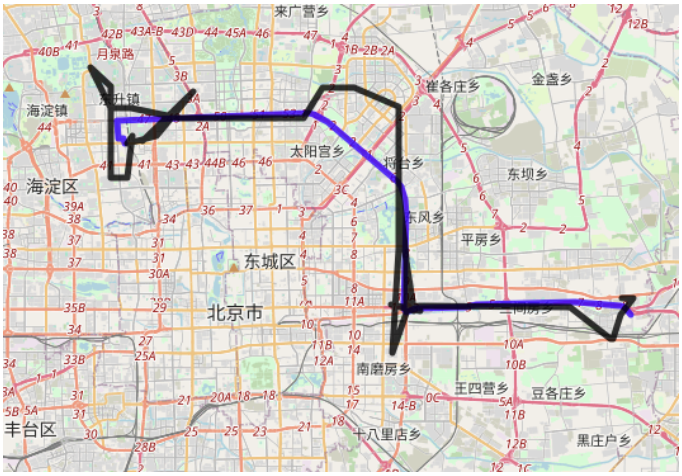


Figure 4.3: An example output after a median filter is applied. Blue: actual trajectory, black: published trajectory de-noised by a median filter.

the noise is added in polar coordinates with angle and distance perturbation, where the angle perturbation is uniformly random, resulting in the output location points around the actual trajectory. As shown in Figures 4.2 and 4.3, when adversaries apply the median filter to de-noise the output trajectory under city road maps, they can achieve the nearly actual trajectory. One approach is to apply noise with similar angle perturbation so that the output trajectory tends to be parallel to the actual, which can lead the adversary to a wrong trajectory. To achieve similar angle perturbation while providing privacy guarantees, differential privacy is considered with the Gaussian mechanism that

$$\theta = \theta_0 + \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{2\ln(1.25/\delta)\cdot(\Delta_2)^2}{\epsilon_a^2}\right). \tag{4.6}$$

where θ is the new angle perturbation considering the previous angle θ_0. The algorithm satisfies (ϵ_a, δ)-differential privacy. Since the algorithm is run for multiple rounds, the composition of differential privacy is needed. After sequentially applying k rounds, the algorithm provides $(O(\sqrt{k})\epsilon_a, \delta')$-differential privacy [7, 35]. The ANGLE procedure in Algorithm 5 shows the angle selection algorithm, and we further discuss the selection of ϵ_a in Section 4.5.

4.3.3. DISTANCE PERTURBATION

We apply the distance perturbation in a similar way to geo-indistinguishability according to Equation 4.3 as introduced in Section 4.2. The DISTANCE procedure in Algorithm 5 shows how distance perturbation works. We further show its performance in terms of utility and privacy in Section 4.5.

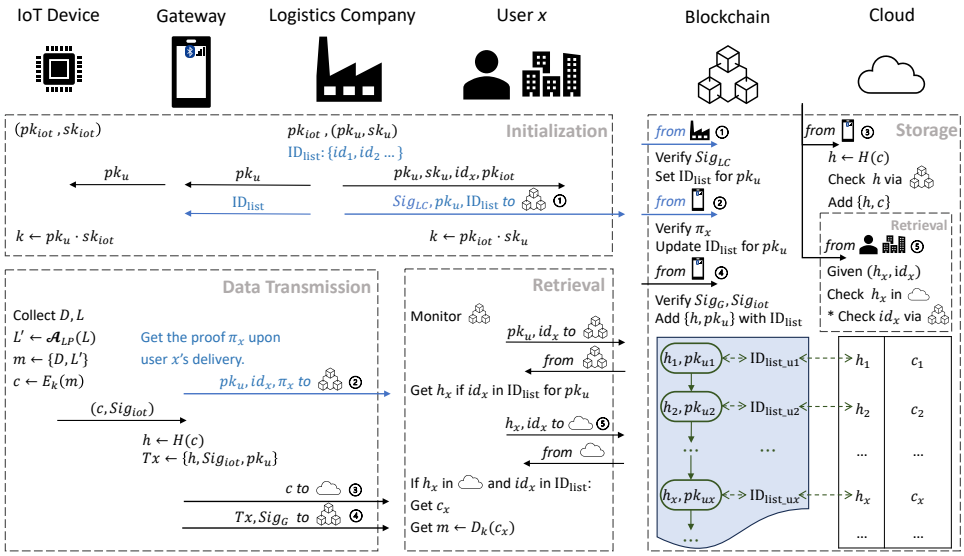


Figure 4.4: Detailed protocols for our design with four phases and six roles. The operations in blue texts consist of the sub-protocol and are run independently between the logistic company and truck drivers for access control.

4.4. SYSTEM ARCHITECTURE

In this section, we present the framework of our design in Figure 4.1. The IoT device is embedded in a container of a truck, where it collects logistics data such as GPS data, temperature, etc. The collected data is encrypted and then transmitted to the gateway via BLE communication. Note that the gateway cannot decrypt the data, and it broadcasts the hash of the encrypted data to the blockchain platform, while the encrypted data itself is stored in the cloud. Users (customers) and logistics companies can access and track their logistics data using the blockchain platform and cloud storage. Notably, data authenticity is achieved via the payload digitally signed by the IoT device. Meanwhile, a sub-protocol is operated by the logistics company and truck drivers (deliverers) to maintain the ID list for access controls in the blockchain and the cloud storage, as shown in blue texts in Figure 4.4. The sub-protocol is run independently of the main protocols (in black texts). Our detailed protocol is shown in Figure 4.4 with four phases: initialization, data transmission, storage and retrieval. In addition, we address how access control works in each phase. We also give detailed privacy and security analysis of our design in Section 4.7.

4.4.1. INITIALIZATION PHASE

In the setup, each IoT device has a unique public-private key pair, and only the public key is shared with the corresponding logistics company. Logistics companies hold the public keys of all their IoT devices, but the companies do not know the private keys. In Figure 4.4, for each transportation, the logistics company first generates a key pair (pk_u, sk_u), and the public key pk_u can be seen as the transportation ID. pk_u is shared

with the IoT device to enable it to start the delivery session. Both (pk_u, sk_u) and pk_{iot} are shared with the customers so that the customer and the IoT device can generate the same shared symmetric key k using the DHKE while keeping their own private key safe. The shared symmetric key can then be used to encrypt and decrypt the logistics data. For access control, the logistics company generates an ID list to include which users have access to the logistics data. Then, it shares the list with the truck driver (gateway) and the blockchain for initialization. After that, the ids are assigned to all the customers for the truck. Note that only the sub-protocol can access the ID list, and the sub-protocol is independent of the main protocols, as shown in Figure 4.4.

4.4.2. DATA TRANSMISSION PHASE

In this phase, as shown in Figure 4.4, the IoT device first collects logistics data, which includes (1) general data D, such as humidity, temperature, etc. and (2) location data L, which is the GPS data of the truck. The location data is sanitized by Algorithm 5 with predefined ϵ' and ϵ_a. The IoT device then uses the symmetric key k to encrypt the sanitized logistics data. After that, the IoT device signs the encrypted data c and sends it to the gateway. The digital signature on the encrypted data ensures authenticity. Hence, only a registered IoT device can issue logistics data for the goods. Moreover, the device is authenticated in distributed settings, thanks to a specific smart contract that allows registering the device's public key. Note that the IoT device, in this PoC, is the source of trust as it is capable of providing location privacy since the data is not issued by a gateway that can be more easily manipulated by a misbehaving entity. Afterwards, the gateway calculates the hash of c and sends the hash (instead of c) to the blockchain. Also, the gateway signs the blockchain transaction to prove the identity. As the transaction is signed by the gateway, we assume that the gateway is known to the blockchain network by its public key. Similarly, the gateway sends c to the cloud storage. For access control, the ID list is shared by the logistics company, and the truck driver collects the delivery proof from the recipients. The delivery proof can be the signature of the recipient and the confirmation of delivery from the logistics company based on the signature. The truck driver then sends the removed id_x to the blockchain with the transportation ID pk_u and delivery proof π_x (note this is part of the sub-protocol).

4.4.3. STORAGE AND RETRIEVAL PHASE

In the blockchain, the smart contract stores the hash value of c under the transportation ID if and only if the signature of the IoT device Sig_{iot} on the hash value of c is valid. Since the validity of the signature is proven in a decentralized manner, the overall system can provide trustworthy data storage management. Moreover, our application for constrained IoT devices allows issuing digital signatures using hardware-secure private keys. As these private keys cannot be accessed and they are unique for each IoT device, the signature creation is considered trusted. We include further details on private keys in Section 4.6.1. For the cloud, when it receives c from the gateway, it computes the hash of the encrypted logistics data. If the hash is included in the blockchain (from a valid source), the cloud adds the hash and the encrypted data to the database. In case the content of c was manipulated, which can be verified via hash validation, the cloud can claim the invalidity of c.

For retrieval, pk_u is used as the index for customers to find their transportation and get the hash of c from the blockchain. With the ID list, only customers in the ID list have access to the hash h. The hash is then used to locate the corresponding logistics data in the cloud. Similarly, the cloud first checks whether h is included in the storage. Then, the cloud checks whether the user id is in the ID list by accessing the smart contract. After the verification, a valid user can access c and decrypt it using the shared symmetric key k to get m. Note that both the blockchain and the cloud do not have k and cannot decrypt c. Meanwhile, when a parcel is delivered, the smart contract updates the ID list by removing id_x upon a valid delivery proof π_x. As a result, the customer does not receive any additional logistic updates for the truck. After the delivery, customers only lose access to further data, and they can still track their old logistics data for their package in our platform. We include further security and privacy analysis for our protocols in Section 4.7.

4.5. EXPERIMENTAL RESULTS FOR TRAJECTORY PERTURBATION

In this section, we introduce our experiments for the trajectory perturbation algorithm, including the selection of privacy parameters ϵ and ϵ_a, and the utility-privacy trade-off evaluation. In this section, the evaluation is based on a laptop with Windows 10 Pro, Intel Core i7-10710U CPU and 16.0 GB RAM. The IoT-related application evaluation and performance are further included in Section 4.6.

For the experiments, we use the GPS trajectory dataset [36], including 1,292,951 kilometres and 17,621 trajectories collected by GPS loggers and phones from 182 users. In the dataset, we consider the trajectories with transportation modes 'Bus' and 'Car & Taxi'. We test our trajectory perturbation algorithm based on the map of Beijing. For simulation, each trajectory is considered as one delivery of the truck.

4.5.1. DISTANCE METRIC

For evaluation, we use the Haversine formula as the distance metric to calculate the distance between two GPS location points, as shown in Equation 4.7. Here φ and λ are the latitude and longitude, and r is the radius of the Earth.

$$d((\varphi_1,\lambda_1),(\varphi_2,\lambda_2)) = 2r \arcsin\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+\cos\varphi_1\cdot\cos\varphi_2\cdot\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}. \quad (4.7)$$

In this section, we utilize $d((\varphi_1,\lambda_1),(\varphi_2,\lambda_2))$ to evaluate the distance between the output and the actual location point (Avg. distance in Table 4.1), and the distance difference between the truck and the destination using output and actual location points (Avg. error in Table 4.1).

4.5.2. PRIVACY PARAMETER SELECTION

ϵ_a

ϵ_a determines the amount of noise added to the perturbed angle, influencing whether the output angle is similar to the previous. According to the Gaussian mechanism, we can calculate the probability $\Pr(\delta_\theta)$ that the output angle $\theta \in [\theta_0-\delta_\theta, \theta_0+\delta_\theta)$ where θ_0 is

the previous angle and $\delta_\theta \in (0, \pi]$. The probability function is shown in Equation 4.8. Δ_2 is the l_2-sensitivity for angle selection, which is 2π in our scenario.

$$\Pr(\delta_\theta) = \frac{\int_{-\delta_\theta}^{\delta_\theta} f(x)dx}{\int_{-\pi}^{\pi} f(x)dx}$$
$$\text{where } f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \ \sigma^2 = \frac{2\ln(1.25/\delta)\cdot\Delta_2^2}{\epsilon_a^2} \tag{4.8}$$

In Figure 4.10, we further illustrate the relation between the value of ϵ_a and the probability of the output angle within the range of $\delta_\theta \in \{1/4\pi, 1/8\pi, 1/12\pi\}$ to the previous angle. With an increasing ϵ_a, the output angle is closer to the previous angle. For example, when $\epsilon_a = 150$, the output angle is nearly 100% within $1/8\pi$ and ~ 90% within $1/12\pi$ to the previous angle. However, the high probability implies a series of similar output angles, which can leak information about how the angle is perturbed. In this paper, we select $\epsilon_a = 20$ to control the output angle. With $\epsilon_a = 20$, the output angle is 49.8% within $1/4\pi$, 26.3% within $1/8\pi$, and 17.7% within $1/12\pi$ to the previous angle so that the output trajectory tends to a similar direction to the actual while preserving location privacy. Under the assumption of differential privacy, the adversary has access to all records in a dataset except one. In our scenario, the adversary can never access any output angles. As a result, though $\epsilon_a = 20$ is a large value for differential privacy, privacy is still preserved and can provide even better privacy guarantees for trajectory perturbation.

4

ϵ

ϵ determines the amount of noise for distance perturbation for geo-indistinguishability. As shown in Table 4.1, the average distance can be used to estimate the distance between the published and the actual location point. With the average distance difference, we can use real maps to consider road density to find whether the real trajectory is hidden among different roads. For example, Figure 4.9 shows the potential distance perturbation of a specific location point with $\epsilon = 0.001$ and $\epsilon = 0.003$. From the figure, the average distance perturbation circle covers 9 main roads and more than 20 different streets. Similarly, when $\epsilon = 0.003$, the average distance perturbation circle is smaller and covers only three main roads. Based on the experimental results, we can select ϵ under road maps. In this paper, we choose $\epsilon = 0.001$.

4.5.3. UTILITY AND PRIVACY EVALUATION

PRIVACY PRESERVATION

In Figures 4.2 and 4.3, we illustrate the output trajectory under possible attacks such as a median filter. Though the perturbed trajectory seems randomized in Figure 4.2, the noise can be well removed using a median filter. Based on Figure 4.3, an adversary can re-identify the actual trajectory of a truck. Similarly, we give further examples in Figures 4.5 and 4.7 to show that a median filter can denoise and re-identify the actual trajectory. On the contrary, Figures 4.6 and 4.8 demonstrate that the output trajectory of $\mathcal{A}_{LP}$ can mislead the adversary to a new trajectory instead of the actual one. With the output trajectory, the adversary will probably infer another road and trajectory which

Table 4.1: Average distance and average distance error with different ϵ.

ϵ	Avg. distance (m)	Avg. error (m)
0.0001	23174.54	20442.49
0.0005	4619.80	3329.57
0.001	2310.57	1583.53
0.003	769.97	528.39
0.005	462.46	318.25
0.007	329.28	226.01
0.01	231.51	156.80
0.05	46.17	29.83

is parallel to the actual road. By misleading the adversaries, we can enhance location privacy protection in practice. Further privacy analysis is included in Section 4.7.

UTILITY EVALUATION

Table 4.1 implies that the publish location point is around 2.3 km away from the actual when $\epsilon = 0.001$. Such deviation does not greatly influence the impact of location data on eCMR for truck location control. Meanwhile, the average distance error indicates the error distance between the truck and the destination. Since the average error is around 1.6 km, the error can lead to an approximate five-minute difference in arrival time estimation. The application of $\mathcal{A}_{LP}$ can maintain the utility of trajectory data for both eCMR and arrival prediction.

4.6. IMPLEMENTATION AND EVALUATION FOR CONSTRAINED IOT DEVICES

In this section, we focus on the technical aspects of implementing the proposed solution. Our PoC contains three main components: the API that is enabled for constrained IoT devices, the gateway application that forwards the messages received from the IoT devices via Bluetooth Low Power (BLE) communication, and the smart contract that includes access policies, signature verification and storage mechanisms. One main aim of the experiments is to validate that our proposed location privacy algorithm is capable of operating efficiently in a constrained IoT device. Another objective is to demonstrate the impact of using the proposed protocol in these devices.

4.6.1. IOT DEVICE API

Our proposed protocol for constrained IoT device applications is developed in C++ and available as open-source code. Our implementation has also been adopted for the Arduino environment, and we outsource transaction generation from the IoT device using the approach in [22]. For the experiments, we use an Arduino Nano 33 IoT device equipped with an ARM Cortex-M0 CPU and an ATECC608A crypto chip [37] for accelerating cryptographic primitives, secure private key storage, and truly random number generation. Our protocol for IoT devices contains four main operations as follows.

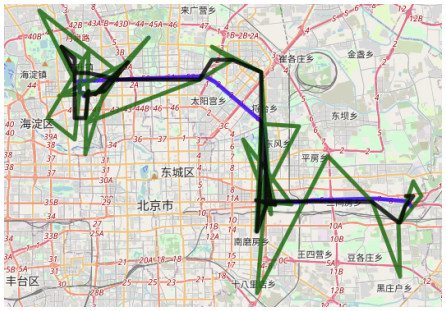


Figure 4.5: An example output by PL_ϵ on a real map with $\epsilon = 0.001$. Blue: actual trajectory, green: published with PL_ϵ, black: under median filter.

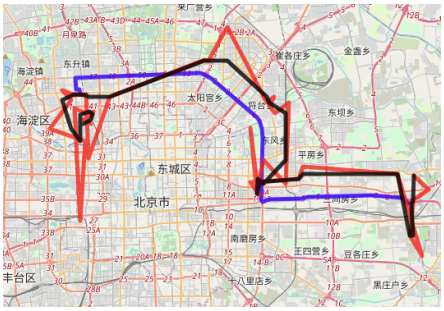


Figure 4.6: An example output by $\mathcal{A}_{LP}$ with $\epsilon = 0.001$. Blue: actual trajectory, green: published with $\mathcal{A}_{LP}$, black: under median filter.

Figure 4.7: Another example output by PL_ϵ on a real map with $\epsilon = 0.001$.

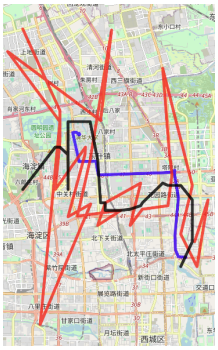


Figure 4.8: Another example output by $\mathcal{A}_{LP}$ with $\epsilon = 0.001$.

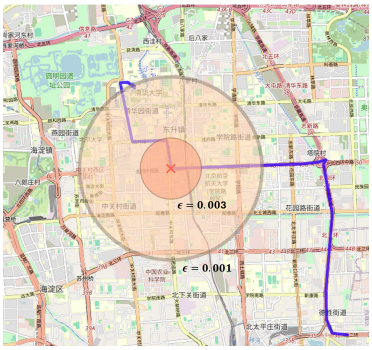


Figure 4.9: Average distance circle from an example location point with $\epsilon = 0.001$ and $\epsilon = 0.003$.

TRAJECTORY PERTURBATION ALGORITHM

The operation corresponds to Algorithm 5. As the algorithm includes random number generation operations, it is essential to note that the truly random number generation in certain embedded devices is still challenging [38]. Therefore, our implementation contains functionalities to ease the usage of an Internal High-Quality NIST SP 800-90A/B/C Random Number Generator (RNG) embedded in the ATECC608A crypto chip.

ENCRYPTION

AES cipher is applied using the symmetric key to encrypt the sanitized data. Our implementation for AES is based on the Tiny AES C library [39] to realize the AES-CBC cipher. One strength of our protocol is due to the frequent update of the symmetric key. Each time when a new session of delivery is initiated, a new symmetric key k is derived using DHKE. For DHKE operation, we employed the Trezor-Crypto library [40] using P-256 NIST curve. Since the DHKE is a computationally expensive operation and the private key storage is challenging in constrained IoT devices, we also propose a hardware-secure DHKE execution in the ATECC608 crypto-chip. Note that the original ArduinoECC08 li-

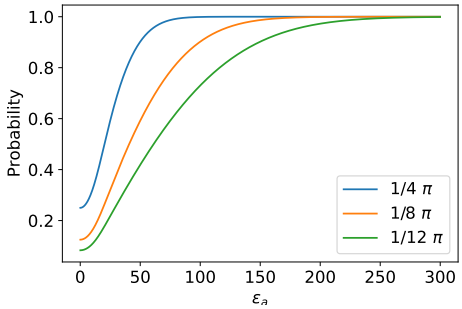


Figure 4.10: Relation between ϵ_a and the probability of an angle output within the range of $n_i \in \{1/4\pi, 1/8\pi, 1/12\pi\}$ to the previous angle.

brary does not contain an interface for DHKE operation. Therefore, we provide a patch for the original library.

SIGNATURE

The signature Sig_{iot} is obtained by employing the ECDSA digital signature algorithm using the P-256 NIST curve. Our software-based implementation inherits the digital signature operations and the necessary SHA-256 cryptographic hash function from the Trezor-Crypto library. We also provide an interface to enable the call of the ECDSA signature operation executed by the ATECC608 crypto-chip. With this partial hardware execution, the signature can be performed faster and more securely as the private key is hardware-secure. Note that the private keys cannot be accessed and modified from the micro-controller that embeds the ATECC608 crypto-chip or to which the chip is connected. In addition, the private keys are generated only once in the initialization phase while programming the crypto-chip. The temper-proof security is guaranteed since only the results of operations requiring the use of the private keys are accessible.

COMMUNICATION (TX/RX)

The communication in our PoC between the IoT device and the gateway is managed via Bluetooth Low Energy (BLE) Generic Attribute (GATT) profile [41]. The IoT device advertises a main service and three characteristics that the gateway subscribes to. The main characteristic is the sending (Tx) of the sanitized data. The remaining characteristics are used for reception purposes (Rx). They indicate if a session of delivery is ended, or if a new session ID (pk_u) must be used to generate a new symmetric encryption key k. In Figure 4.11, we denote these reading characteristics as “Notification flag" and “session ID flag", respectively. In our protocol, the Bluetooth radio module is switched off before and after the Tx/Rx communication. Furthermore, each Tx is followed by a short listening phase (Rx) to see if the session has ended, meaning the device can remain in a power-safe “Sleep mode" for a longer period of time. We use the ArduinoBLE library [42] to establish the BLE communication between the IoT device and the gateway.

Since energy consumption is crucial in wireless sensor networks, as most devices are powered by batteries with a limited lifetime, the design of processing states and the

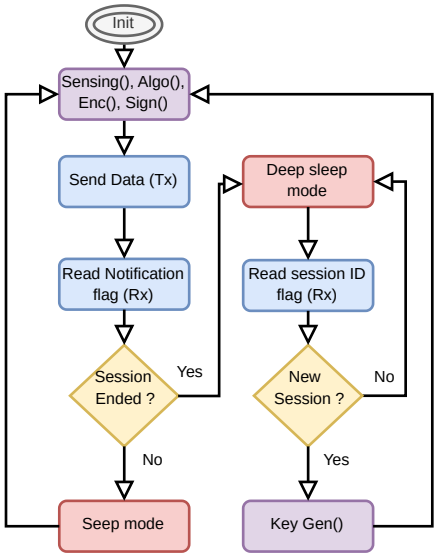


Figure 4.11: Processing states in the IoT application.

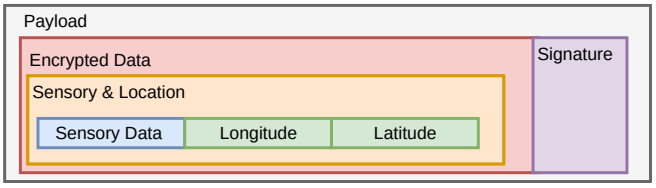


Figure 4.12: The Protobuf Payload.

limitation of the number of communications must be precisely planned. Figures 4.11 and 4.12 shows the processing states of our protocol and the payload structure used. After the initialization phase, the device can perform sensing operations specific to the given use case (e.g., temperature, humidity, etc.). Following the sensing operations, the device runs the privacy-preserving location algorithm. The result of the algorithm and the sensed data are incorporated into a specific structure. Afterwards, the sanitized data is encrypted and digitally signed. It is worth noting that the payload, which contains the encrypted data and its signature, is realized using Protobuf [43] data format that allows a language-neutral serialization of the structured data. The serialized data structure can then be sent to the gateway via BLE (Tx state). Then, the “Notification flag" is read to identify if the session is ended or not. In case the session continues, the device passes to an energy-saving “Sleep mode". In our context, the sleep mode is fixed to 5 minutes. After the sleep time has elapsed, the device comes again to the sensing state, and the previously described processing states repeat once again. In case the session is ended,

Table 4.2: Runtime energy consumption and execution time of our protocol. The table also compares the measured metrics when hardware accelerations are applied for the Elliptic Curve Digital Signature, the Elliptic Curve Diffie-Hellman and random number generation. Note: Total* energy consumption does not include the Key Generation because this operation is called only when a new session of the protocol starts.

Mode	Metric	Algorithm	Encryption	Signature	Key Gen.	Total*
Pure Software	Avg. ΔE	0.4mJ	62.1μJ	85.3mJ	360mJ	85.8mJ
	Avg. ΔT	3.6ms	540μs	741.4ms	3.13s	745.5ms
Partial Hardware	Avg. ΔE	9mJ	62.1μJ	17mJ	17.2mJ	26.1mJ
	Avg. ΔT	76.6ms	540μs	123.3ms	137ms	200.4ms

the device proceeds to a “Deep sleep mode", which can be a greatly longer period, such as 1-2 hours. Waking up from the “Deep sleep mode", the next step is to verify if a new session ID (pk_u) was provided by the gateway. If it is not present, the device returns to the “Deep sleep mode". Otherwise, a new symmetric key (k) is generated using the new session ID, and the device can move to the sensing state.

In regards to secure communication between the IoT device and the gateway, the authentication protocol can vary depending on the communication protocol used. Nevertheless, authentication of the IoT and the gateway can be ensured by applying TLS or other authenticated encryption protocols on top of the given communication protocol. As our study focuses on the feasibility of trajectory tracking in constrained IoT devices, we left out authenticating the IoT device to the gateway.

4.6.2. GATEWAY API AND SMART CONTRACT

The gateway API acts as a communication bridge between the IoT device, the blockchain and the cloud. Our proposed gateway API allows the connection of an IoT device by specifying the BLE MAC address of the device. The API is supplied with the official fabric-go-sdk [44] to enable the generation of valid blockchain transactions and interactions with smart contracts. In this paper, we consider Hyperledger Fabric as the blockchain platform, and we further introduce the details of blockchain applications in Section 4.6.5. When the IoT device is in sleep mode, the connection is also stopped. Therefore, the gateway API can be turned into a re-scanning phase to connect the IoT device again. The gateway API has two main functionalities. After the serialized Protobuf BLE packets (sent by the IoT devices) are received and decompressed, the encrypted data (c) and its hash value are forwarded to the cloud back-end service, while the hash value and the signature are embedded into a blockchain transaction and sent to a specific smart contract. In addition to forwarding BLE data to the blockchain, the gateway API also informs the IoT device about the end of the delivery session and the opening of a new session by providing a new session identifier pk_u). Our proposed gateway API is implemented in Golang, and it uses the Go Bluetooth [45] cross-platform package for BLE. Our work is tested on a Linux operating system.

The proposed smart contract in Hyperledger Fabric, after receiving the transaction created by the gateway, first verifies if the signature on the payload issued by a registered device is valid. We assume the public key (pk_{iot}) of the IoT device is registered at the smart contract by the logistics company. The data storage in the distributed ledger is

Table 4.3: Estimated days of autonomy when our protocol is or is not applied. The results also show the different battery life when only the software implementation (SW) and the partial hardware acceleration (HW) are applied.

	Without our protocol	With our protocol (SW)	With our protocol (HW)
ΔE static (J)	22.29	22.29	22.29
ΔE dynamic (J)	1.83	1.92	1.856
ΔE total (J)	24.12	24.21	24.14
Estimated battery life	641.1h / 26.7day	640.3h / 26.6day	641h / 26.7day

handled via the storage of key:value pairs. If the signature is valid, a composite key is generated by combining the session ID (pk_u) and the hash value of the encrypted data (c), with the corresponding value equal to the hash digest.

4.6.3. IOT EXPERIMENTAL RESULTS

We program the Arduino using Arduino IDE v1.8.19. The principal objectives of the experiments are as follows.

- Determine the power consumption of our proposed protocol with and without applying a partial hardware-secure execution.
- Highlight the impact of our protocol and the location privacy algorithm on the overall execution time and energy consumption of the IoT device.
- Estimate the battery life with and without applying our protocol.

ENERGY CONSUMMATION MEASUREMENTS

In the following measurements, we analyze the energy consumed by different operation states, i.e., location privacy algorithm, encryption, signature, key generation and communication (Tx/Rx), as we previously described. We used the Otii Ace Pro [46] power supply and analyzer to measure the energy consumption and to highlight the consumed energy and time taken by the different operations of our protocol. Table 4.2 represents the average execution time ΔT and the energy consumption ΔE of our proposed protocol. Note that this measurement does not include the energy consumption and execution time of the communication (Tx/Rx) protocol, since the choice of the communication can have a high impact on the overall energy consumption, and other more energy-efficient wide-area network wireless communication protocols can also be embedded into our proposed implementation according to requirements of the given use case.

In the measurements, the sanitized logistic data is constructed from the sensing data, which we have fixed to 5 bytes but whose length can vary up to 1024 bytes and the coordinates (longitude, latitude) are represented by two 32-bit floating numbers. The results also highlight the performance differences between applying hardware-secure computing (HW) on cryptography-related operations and executing the pure software implementation (SW) of the protocol.

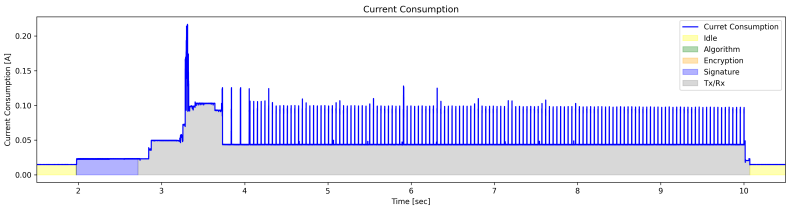


Figure 4.13: Current consumption of the protocol while sending a digitally signed location information protected with our location privacy algorithm

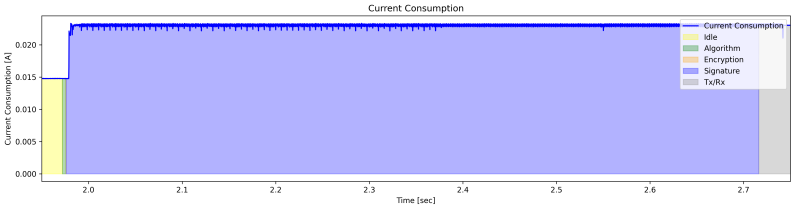


Figure 4.14: A closer look on the current consumption of our protocol when the Tx/Rx communication is out of scope

After observing the results of Table 4.2, we can conclude that the implementations of the location privacy algorithm are competitive with other operations in the protocol. Nevertheless, there are significant differences in execution time and power consumption between the software implementation and the hardware-secure location privacy algorithm. This is due to the computation time of random numbers generation by the internal NIST 800-90 A/B/C random number generator, and the communication overhead between the crypto chip and the Arduino board. Overall, the software implementation of the location privacy algorithm can be executed in 3.6ms, while the same algorithm using a NIST-recommended random number generation takes a considerable 76.6ms.

We notice that the Encryption operation takes the shortest execution time among other operations. However, the encryption depends on the size of the plaintext, i.e., the larger the sensing data, the longer the execution time. In this setting, the plaintext size was relatively small. Meanwhile, the signature and the key generation have fixed execution times, as the input data used by these algorithms are fixed. We can also observe that the signature and key generation operations can be highly accelerated, and more efficient energy consumption can be achieved by using the hardware acceleration provided by the ATECC608A crypto chip.

Figure 4.13 illustrates the current consumption of a computation period of our protocol and the communication phase via BLE. The same figure also highlights the current consumption of the sleep mode that we display as the “Idle” phase. As we can see, the communication phase takes significantly longer time than the proposed protocol. Figure 4.14 shows the current consumption of the different operations of our protocol without the communication phase. The retrieved current consumption highlights the execution of the software implementation.

AUTONOMY ESTIMATION

In this experiment, we estimate the achievable autonomy by our PoC. Therefore, the time and energy consumption of the communication phase (Tx/Rx) is taken into account. We assume that the device is supplied by four 3.4Ah batteries that are considered perfect, and they deliver 3.7V until complete discharge, and the energy consumption of the voltage regulator between the batteries and the device is negligible. Table 4.3 highlights the estimated autonomy of the IoT device. The total energy consumption is the sum of the static and the dynamic energy consumed. In the following, we assume that the device data sending rate is 300 seconds (5 minutes), which also means that the device is in sleep mode for 5 minutes, in this mode the current consumption is 14.7mA ("Idle" state), and the reciprocal energy consumption is 22.29J. In the scenario when our protocol is used, the dynamic energy consumption is the sum of the consumption of our protocol (as in Table 4.2) and the BLE communication, which is 1.83J. Obviously, when our protocol is not applied, the dynamic energy consumption is equal to the energy consumption of the BLE communication.

Estimations are calculated with the following equations:

$$\begin{aligned} Estimation(hours) &= \frac{E_{battery}(Wh)}{E_{total}(W)}, \\ E_{total}(W) &= \frac{E_{total}(J)}{Period(s)}. \end{aligned} \tag{4.9}$$

The estimated battery life shows that our proposed protocol only slightly impacts total energy consumption since, without the protocol, only one hour of autonomy is gained compared with the purely software-based implementation. When we apply hardware acceleration on cryptographic-related operations, the values estimated for the battery life with and without using our protocol are almost equal.

Note that in our evaluation, the data is sent from the IoT device to a gateway module via BLE. Other low-power energy consumption communication protocols, such as LoRaWAN [47], allow more efficient energy consumption with a shorter time delay. The LoRaWAN configuration [22] is nearly 6 times faster than the BLE communication when the data is less than 242 bytes long. Note that the Arduino board used in our experiments is a "Low-Power" architecture instead of "Ultra-Low-Power", meaning that its consumption in the "Idle" phase is around 14 mA, which is largely above the "Ultra-Low-Power" characteristic with approximately 100 μA. The application of an "Ultra-Low-Power" board implies a longer battery life, and the proposed location privacy algorithm itself is efficient in terms of execution time and energy consumption. It is feasible to employ our approach in other constrained embedded systems and IoT devices with a slight impact on the overall consumption of the given device.

4.6.4. TRAJECTORY PERTURBATION ON CONSTRAINED IOT DEVICES

We implement Algorithm 5 in C++ and compare the output trajectory to [7] (with Python) in Figures 4.15 and 4.16. The performance of our implementation is at the same privacy level as shown in Section 4.5 with the same $\epsilon = 0.001$. Note that the result cannot be the same since the random values are different for each run, but the level of noise is similar. Meanwhile, the output location point is around a block away from the actual, which sup-

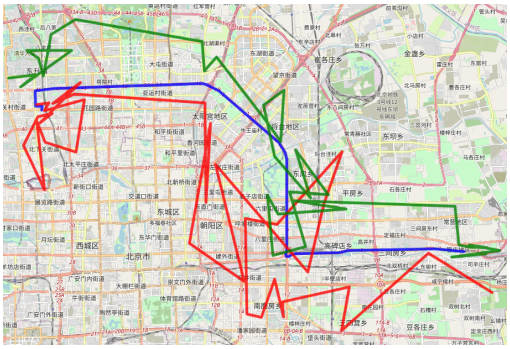


Figure 4.15: Example output trajectory with actual in blue, with C++ (IoT enabled) implementation in green, and with Python (for generic use in [7]) implementation in red.

ports the idea that it can protect the actual location of trucks in [7, 13]. Meanwhile, the randomness of the algorithm is guaranteed by the NIST-recommended random number generator within the IoT device.

Note that the random number generation in our implementation is also feasible by calling Arduino's random number generation features, which is faster than calling the NIST-recommended crypto chip we presented. However, Arduino's random number generation uses a fairly random input for the seed instead of truly random. The weaker guarantee of randomness can lower the privacy guarantee of our trajectory perturbation algorithm $\mathcal{A}_{LP}$. Also, there remains more work for truly random number generation on constrained devices.

4.6.5. IOT-BLOCKCHAIN APPLICATIONS

In our implementation, we rely on Hyperledger Fabric, which is a permissioned blockchain that allows the deployment of sophisticated access policies. Also, the new member registration to the blockchain network is restricted. With the access control, only authorized users can add or read transactions on the ledger. Another advantage is its modular architecture, allowing the usage of different cryptographic signature schemes. Moreover, Hyperledger Fabric eases the deployment of smart contracts since they can be implemented in different programming languages. Thanks to its modularity, easy policy-making, and popularity in enterprise-level use cases, Hyperledger Fabric is a promising candidate for supply chain and logistics applications. To showcase its performance in real-life use cases, we refer to the study of Guggenberger *et al.* [48], indicating that a Hyperledger Fabric network deployed in four different European countries and containing a total of eight peer nodes can achieve a throughput of over 1000 transactions per second with a latency of 1.2 seconds. According to these results, we can confirm that our IoT-blockchain-based location privacy platform is suitable to be implemented in practice since the overall latency of a location data sending, including its distributed validation and storage, would not exceed more than 1.5 seconds.

Regarding real-world applications of IoT-blockchain implementations, the IoTeX project [49] uses hardware secure computation to ensure the root of trust in IoT-blockchain

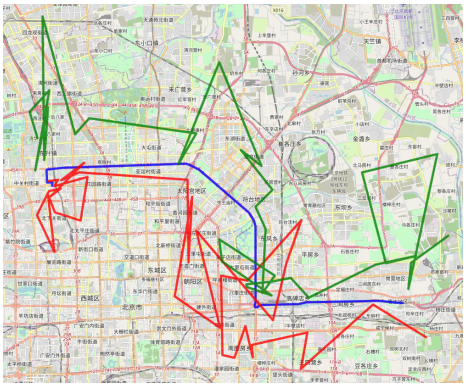


Figure 4.16: Example output trajectory with actual in blue, with C++ (IoT enabled) implementation in green, and with Python (for [7]) implementation in red.

communication. However, IoTeX-issued constrained IoT devices contain an ARM CryptoCell-310 crypto module [50], which is not scalable since it is only compatible with Arm Cortex M processors, which can only be found as an integrated module in SoCs. In our work, we apply the ATECC608A cryptographic chip, which is compatible with a wide variety of constrained IoT devices. In addition, IoTeX requires the use of the IoTeX public blockchain, which implies an amount of gas price to be paid after each transaction issued. Our proposed solution is considered more generic, as it is compatible with any kind of blockchain because the payload of the transaction is signed locally in the device, which can be afterwards verified in any kind of smart contract.

4.7. ANALYSIS

4.7.1. TRAJECTORY PERTURBATION

In Section 4.5, we illustrate how privacy parameters are selected and the performance of our trajectory perturbation algorithm in terms of privacy protection and utility for eCMR and arrival prediction. In theory, the angle perturbation algorithm also provides stronger privacy guarantees than uniformly random perturbation [7]. The distance perturbation provides strong privacy guarantees based on geo-indistinguishability [13]. Meanwhile, we optimize existing solutions to be more adaptable for deployment on constrained IoT devices with lower computation and storage costs.

In the work of [7], different privacy parameters are considered, resulting in N encryption using the public key of N different users to encrypt n AES keys, where the n AES keys are used for the encryption of n different location outputs and n is dependent on the variety of privacy parameters. As a result, $n+N$ encryption is needed. With $\mathcal{A}_{LP}$ in our paper, we optimize the privacy parameter selection procedure. Only one message is encrypted and transmitted to the blockchain since different users share the AES key for the encryption of the perturbed location data. In theory, we expect that $\mathcal{A}_{LP}$ only consumes approximately $1/(N+n)$ costs in terms of time and energy for encryption.

4.7.2. SYSTEM SECURITY AND PRIVACY ANALYSIS

In Figure 4.4, we assume the key generation scheme, key exchange scheme, symmetric encryption scheme, signature scheme, and hash functions are secure by design. We consider the security and privacy of our protocol in terms of confidentiality, integrity and availability in Theorems 4.1 to 4.5.

Lemma 4.1. *A probabilistic polynomial time (PPT) adversary $\mathcal{A}$ cannot access or change the transmitted logistic information.*

Proof. Our adversary model assumes that the gateway is malicious. Only the IoT device can create the logistic data and transmit it to the storage via the gateway. The user and the logistics company can access their logistic data.

The logistic data is encrypted by a shared key between the IoT device and the user. Note that the gateway (the PPT adversary $\mathcal{A}$) only holds pk_u, so it cannot generate k. Based on the security of the key exchange scheme and the symmetric key encryption, it is computationally infeasible for $\mathcal{A}$ to decrypt c without the symmetric key. $\mathcal{A}$ cannot access the logistic information.

Similarly, based on the security of the signature scheme, the gateway, the user and the logistics company ($\mathcal{A}$) cannot change c or transmit fake encryption c' to the storage since this cannot pass the signature verification from the smart contract. Note that the logistics company cannot forge the signature or private key of the IoT device when our protocol is used with the hardware secure module (with the assumption that the hardware module does not reveal any information about the private key, since it cannot be accessed). However, in our software-based solution, the private key of the IoT device must be issued by an independent third party that does not collude with the logistics company. The cloud then rejects the addition to the database since the hash is not included in the blockchain. $\mathcal{A}$ cannot change the logistic information.

Meanwhile, based on the security of the signature scheme and the verification with the smart contract, the proposed protocols can protect against possible attacks such as MITM attacks or spoofing. □

Lemma 4.2. *The system can provide trustworthy data authenticity and validation with the blockchain against PPT adversaries.*

Proof. With the application of blockchain, data authenticity and validation can be achieved in a decentralized manner. Considering a PPT adversary, the signatures of the gateway and the IoT device are verified using the smart contract. Also, data validity is based on the decentralized verification implemented in the smart contract. The trusted data validation process holds as long as the consensus process is valid, implying that the majority of the consensus rule participants are honest, such as the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm requiring that 2/3 of the participants be honest. □

Lemma 4.3. *Logistic data must arrive on the blockchain periodically within a predetermined time error margin. If the adversary $\mathcal{A}$ denies the message forwarding, the system can detect it and notify the users.*

Proof. From our adversary model, we assume that the gateway is malicious. Malicious gateways can possibly deny the transmission of data (by turning off the mobile phone). Although this behaviour cannot be protected by design, our protocol can detect it. We assume that at the beginning of the protocol, a specific time error margin is determined, accounting for expected communication costs and delays. If the waiting time exceeds the margin limit, the blockchain can report the denial of data transmission. Additionally, the protocol allows for setting a threshold on the maximum number of consecutive messages that may fail to appear on the blockchain ledger. If the threshold is exceeded, the blockchain can notify the users of suspicious behaviour during the delivery of the IoT message forwarding process. This may cause users to miss some logistic information and suspect some malicious behaviour. Nevertheless, the received data is always valid and correct according to Theorem 4.2. □

Lemma 4.4. *Only authorized users in the ID list can track their items in the truck. If an adversary $\mathcal{A}$ maliciously forwards an ID list with unauthorized customers, the system can detect it and be known by logistics companies.*

Proof. Based on our assumptions in Section 4.1, when a package is delivered, the truck driver registers the delivery upon proof of the delivery, such as a signature. Meanwhile, as introduced in Section 4.4, the gateway can only remove user ids from the ID list upon the proof of delivery for the package. As a result, the adversary $\mathcal{A}$ do not have access to add new unauthorized users to the ID list or remove the users whose packages are not delivered. Also, $\mathcal{A}$ can not retain existing users whose packages are delivered since the removed id is automatically forwarded to the blockchain upon a valid delivery proof. If $\mathcal{A}$ bypasses logistics companies and sends a new ID list, the blockchain and the cloud can check whether the logistics company verifies the ID list and notify the logistics company if the ID list is forged. □

Remark 4.2. Note that location privacy is aimed at protecting the privacy of truck drivers. In practice, if the truck driver is malicious and wants to share the location of the truck, it is not possible to protect against it since the truck driver can always positively share his location with others.

Lemma 4.5. *All authorized users can access their logistic data validated and correct before their package is delivered. After the delivery, users can not have further information about the truck but can always track and validate their past logistic data.*

Proof. Based on the key exchange scheme, the user (and the logistics company) holds the same shared symmetric key k as the IoT device, so the user can decrypt c and get the logistic information. From the storage, the signatures of the IoT device and the gateway are verified based on the signature scheme, which guarantees that the received logistic information is valid. Meanwhile, Theorem 4.1 shows that $\mathcal{A}$ cannot change the transmitted logistic information, so it is correctly generated by the IoT device. When the package is delivered, the user id is removed from the ID list. Based on Theorem 4.4 and the access control by the blockchain and the cloud, the user can not know whether there is further information for the truck. The user no longer has access to the new blocks in the blockchain nor access to the newly added record in the cloud. For past logistic information, the ID list remains as the old, so the user can still track and validate it. □

Remark 4.3. With our protocol, a third party can validate a series of logistic data without knowing the content. For example, assume a third party wants to buy the logistic data from a user. With the blockchain, the user can use the hash to show the validity of the data without revealing the content. With the transportation ID pk_u, the user can also prove where and when the package is transported.

4.8. CONCLUSION

With our innovative solution, **PrivTrack**, location privacy with constrained IoT devices is now available to track the trajectory of trucks for delivery in big cities. We first show that our trajectory perturbation algorithm provides privacy protection against median filter attacks under real road maps by misleading the adversary to a wrong trajectory. The output perturbed trajectory data is useful for eCMR and package tracking. Also, our evaluation illustrates the feasibility of applying our trajectory perturbation algorithm in constrained IoT devices. Then, we propose a platform that enables the interaction of constrained devices with a blockchain-based platform, offering *data validation*, *authenticity* and *access control*. Our security and privacy analysis show that the proposed protocols provide privacy-preserving data sharing together with the trajectory perturbation algorithm. Based on our experiments, we demonstrate that our protocols have a low impact on the IoT device battery life. When using an IoT device based on an Arm Cortex-M0 processor, the run time is also feasible: **3.6 ms** when the random numbers are generated in software, and **76.6 ms** when generated in a NIST-recommended hardware module. Considering the overall performance of constrained IoT devices, the run time of our protocols is ~200 ms with an autonomy of almost one month, which is well aligned with the needs of real-world use cases.

REFERENCES

[1] J. C. Ferrer. "The CMR Convention - A Pillar of International Carriage of Goods by Road (Abstract)". In: *Uniform Law Review* 11.3 (2006), pp. 521–521. ISSN: 1124-3694. eprint: https://academic.oup.com/ulr/article-pdf/11/3/521/4717296/11-3-521.pdf.

[2] P. Zhang, X. Pang, N. Kumar, G. S. Aujla, and H. Cao. "A Reliable Data-Transmission Mechanism Using Blockchain in Edge Computing Scenarios". In: *IEEE Internet Things J.* 9.16 (2022), pp. 14228–14236.

[3] J. Nadeau. *Banking and Finance Data Breaches: Costs, Risks and More To Know.* 2021.

[4] T. Li, J. Vos, and Z. Erkin. "Decentralized Private Freight Declaration & Tracking with Data Validation". In: *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom 2022 Workshops, Pisa, Italy, March 21-25, 2022*. IEEE, 2022, pp. 267–272.

[5] R. Kromes, T. Li, M. Bouillion, T. E. Güler, V. van der Hulst, and Z. Erkin. "Fear of Missing Out: Constrained Trial of Blockchain in Supply Chain". In: *Sensors* 24.3 (2024), p. 986.

[6] T. Bocek, B. Rodrigues, T. Strasser, and B. Stiller. "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain". In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, May 8-12, 2017*. IEEE, 2017, pp. 772–777.

[7] T. Li, L. Xu, Z. Erkin, and R. L. Lagendijk. "Trajectory Hiding and Sharing for Supply Chains with Differential Privacy". In: *Computer Security - ESORICS 2023 - 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25-29, 2023, Proceedings, Part II*. Vol. 14345. Lecture Notes in Computer Science. Springer, 2023, pp. 297–317.

[8] M. el Maouchi, O. Ersoy, and Z. Erkin. "DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*. ACM, 2019, pp. 364–373.

[9] H. Wu, Z. Li, B. King, Z. B. Miled, J. Wassick, and J. Tazelaar. "A Distributed Ledger for Supply Chain Physical Distribution Visibility". In: *Inf.* 8.4 (2017), p. 137.

[10] M. Grmiling. *How Real Time Tracking Can Improve Logistics*. https://www.hublock.io/how-real-time-tracking-can-improve-logistics/. Accessed: 2021-11-17. 2021.

[11] DHL. *Parcel delivery in real time*. https://www.dhl.de/en/privatkunden/pakete-empfangen/sendungen-verfolgen/live-tracking.html. Accessed: 2022-01-07. 2021.

[12] E. Murati and M. Henkoja. "Location data privacy on MaaS under GDPR". In: *Eur. J. Privacy L. & Tech.* (2019), p. 115.

[13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. "Geo-indistinguishability: differential privacy for location-based systems". In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. ACM, 2013, pp. 901–914.

[14] C. Fang and E. Chang. "Differential privacy with δ-neighbourhood for spatial and dynamic datasets". In: *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014*. ACM, 2014, pp. 159–170.

[15] N. Harnsamut, J. Natwichai, and S. Riyana. "Privacy Preservation for Trajectory Data Publishing by Look-Up Table Generalization". In: *Databases Theory and Applications - 29th Australasian Database Conference, ADC 2018, Gold Coast, QLD, Australia, May 24-27, 2018, Proceedings*. Vol. 10837. Lecture Notes in Computer Science. Springer, 2018, pp. 15–27.

[16] S. Hayashida, D. Amagata, T. Hara, and X. Xie. "Dummy Generation Based on User-Movement Estimation for Location Privacy Protection". In: *IEEE Access* 6 (2018), pp. 22958–22969.

[17] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos. "Local Suppression and Splitting Techniques for Privacy Preserving Publication of Trajectories". In: *IEEE Trans. Knowl. Data Eng.* 29.7 (2017), pp. 1466–1479.

[18] Y. Xiao and L. Xiong. "Protecting Locations with Differential Privacy under Temporal Correlations". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*. ACM, 2015, pp. 1298–1309.

[19] P. Bethi, S. Pathipati, and A. P. "Stealthy GPS Spoofing: Spoofer Systems, Spoofing Techniques and Strategies". In: *2020 IEEE 17th India Council International Conference (INDICON)*. 2020, pp. 1–7.

[20] F. Azzedin and M. Ghaleb. "Internet-of-Things and Information Fusion: Trust Perspective Survey". In: *Sensors* 19.8 (2019). ISSN: 1424-8220.

[21] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani. "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". In: *IEEE Commun. Surv. Tutorials* 21.3 (2019), pp. 2702–2733.

[22] M. Arnaudo, L. Gerrits, I. Grishkov, R. Kromes, and F. Verdier. "Blockchains Accesses for Low-Power Embedded Devices using LoRaWAN". In: *Proceedings of the 12th International Conference on the Internet of Things, IoT 2022, Delft, The Netherlands, November 7-10, 2022*. ACM, 2022, pp. 119–126.

[23] M. Pincheira, M. Vecchio, R. Giaffreda, and S. S. Kanhere. "Cost-effective IoT devices as trustworthy data sources for a blockchain-based water management system in precision agriculture". In: *Comput. Electron. Agric.* 180 (2021), p. 105889.

[24] C. Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35908-1.

[25] C. Dwork, F. McSherry, K. Nissim, and A. Smith. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[26] C. Dwork and A. Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.

[27] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. “Differentially Private Event Sequences over Infinite Streams”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1155–1166.

[28] Q. Xiao, J. Chen, L. Yu, H. Li, H. Zhu, M. Li, and K. Ren. “POSTER: LocMask: A Location Privacy Protection Framework in Android System”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*. ACM, 2014, pp. 1526–1528.

[29] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao. “LocLok: Location Cloaking with Differential Privacy via Hidden Markov Model”. In: *Proc. VLDB Endow.* 10.12 (2017), pp. 1901–1904.

[30] I. Natgunanathan, A. Mehmood, Y. Xiang, L. Gao, and S. Yu. “Location Privacy Protection in Smart Health Care System”. In: *IEEE Internet Things J.* 6.2 (2019), pp. 3055–3069.

[31] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, H. Yu, and D. Liao. “Efficient location privacy algorithm for Internet of Things (IoT) services and applications”. In: *J. Netw. Comput. Appl.* 89 (2017), pp. 3–13.

[32] C. Kjærgaard-Winther. *Maersk and IBM to discontinue TradeLens, a blockchain-enabled global trade platform*. Maersk website. 2023.

[33] A. Sristy. *Blockchain in the food supply chain - What does the future look like?* Walmart website. 2024.

[34] L. Gerrits, R. Kromes, and F. Verdier. “A True Decentralized Implementation Based on IoT and Blockchain: a Vehicle Accident Use Case”. In: *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. 2020, pp. 1–6.

[35] P. Kairouz, S. Oh, and P. Viswanath. “The Composition Theorem for Differential Privacy”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1376–1385.

[36] Y. Zheng, L. Zhang, X. Xie, and W. Ma. “Mining interesting locations and travel sequences from GPS trajectories”. In: *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*. ACM, 2009, pp. 791–800.

[37] Microchip. *ATECC608A Pre-provisioned secure element for TLS system*. https://www.microchip.com/en-us/product/ATECC608A-TFLXTLS#document-table. 2022.

[38] K. Seyhan and S. Akleylek. “Classification of random number generator applications in IoT: A comprehensive taxonomy”. In: *Journal of Information Security and Applications* 71 (2022), p. 103365. ISSN: 2214-2126.

[39] Kokke. *Tiny AES in C*. https://github.com/kokke/tiny-AES-c. 2022.

[40] Trezor-crypto. *Heavily optimized cryptography algorithms for embedded devices*. https://github.com/trezor/trezor-crypto. 2022.

[41] F. J. Dian, A. Yousefi, and S. Lim. "A practical study on Bluetooth Low Energy (BLE) throughput". In: *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2018, pp. 768–771.

[42] Arduino. *ArduinoBLE library*. https://www.arduino.cc/reference/en/libraries/arduinoble/. 2022.

[43] nanopb. *Nanopb - Protocol Buffers for Embedded Systems*. https://github.com/nanopb/nanopb. 2022.

[44] Hyperledger. *Hyperledger Fabric Client SDK for Go*. https://github.com/hyperledger/fabric-sdk-go. 2022.

[45] Bluetooth-Go. *Go Bluetooth cross-platform Bluetooth module*. https://pkg.go.dev/tinygo.org/x/bluetooth. 2022.

[46] Quoitech. *Otii Ace Pro*. https://www.qoitech.com/otii-ace/. 2022.

[47] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke. "A Survey of LoRaWAN for IoT: From Technology to Application". In: *Sensors* 18.11 (2018). ISSN: 1424-8220.

[48] T. Guggenberger, J. Sedlmeir, G. Fridgen, and A. Luckow. "An in-depth investigation of the performance characteristics of Hyperledger Fabric". In: *Comput. Ind. Eng.* 173 (2022), p. 108716.

[49] IoTeX developers. *IoTeX*. https://iotex.io/. 2023.

[50] Arm Developers. *CryptoCell-310*. https://developer.arm.com/Processors/CryptoCell-310. 2023.

5

DATA PRIVACY ENHANCEMENT FOR COLLABORATIVE LEARNING

Automated fraud detection is hindered by a lack of fraud samples in collected data. Collaborative learning can solve this problem by allowing institutions to jointly train a model without directly sharing their data. However, the applicability of collaborative learning is restricted due to its vulnerability to inference attacks, which aim to extract sensitive information from the model. Such attacks are effective in a decentralized, sequential collaborative learning setting where participants have white-box access to the model and can view a portion of the training data, posing privacy risks and deterring collaboration. To address these challenges, this paper introduces a novel framework and protocols that leverage secure multi-party computation and differential privacy to mitigate inference attacks and side-channel timing attacks in a small-scale sequential collaborative learning setting. The protocol begins with participants collaboratively establishing the training order. Together with an anonymous communication network, each participant knows only the destination for their data transmission without knowledge of the source. Additionally, responder anonymity is achieved through the use of Tor's hidden services, which conceal the identities of recipients. To the best of our knowledge, our work is the first to mitigate inference attacks using a secure joint permutation selection protocol with a low overhead of a few seconds. Besides, we apply differential privacy during the training process against membership inference attacks. Moreover, we introduce proper time delays based on differential privacy to mitigate side-channel timing attacks under anonymous communication networks. Our detailed privacy and security analysis illustrates that our framework and protocols enhance privacy protection against side-channel timing attacks and inference attacks.

This chapter is a copy of the paper titled "Robust Small-Scale Collaborative Learning Against Inference Attacks" by Li, T., van Tetering, D., and Erkin, Z., which is under review from *IEEE Transactions on Information Forensics and Security*.

5.1. INTRODUCTION

Automated fraud detection is of great importance with advances in e-commerce. For personal, according to the European Central Bank, billions of Euros are lost due to credit card fraud every year [1]. For business, it is essential to manage financial fraud for supply chain risk management [2]. The automated fraud detection system is widely used to audit all transactions. Financial institutions and companies create such systems by training machine learning algorithms on transaction data. However, the performance is obstructed by a lack of positive (fraud) samples [3]. An ideal solution is to merge the data of all institutions or companies and train on the resulting dataset. Unfortunately, the included sensitive information and security risks for transferring render this solution unrealistic [4]. Moreover, laws such as GDPR in Europe prohibit the sharing of original sensitive data [5]. This motivates the need for protocols that allow collaborative training without sharing private data and comply with GDPR.

Collaborative learning deals with the lack and leakage of training data while following GDPR [4]. It allows parties to jointly train a model in *parallel* or *sequential* settings without sharing data [6]. In the parallel setting, each party trains the model locally and then shares the model parameters with a central entity that aggregates the parameters to get the final model [7]. In contrast, a central entity is not needed for the sequential setting where every participant trains the same model one after the other [6]. Recently, sequential collaborative learning has garnered attention within the collaborative learning community [8, 9]. It shows benefits in terms of fewer training rounds [10] and has advantages when each party holds small datasets [11], which are essential for collaborative learning. Also, it is vital for other emerging distributed learning techniques such as split learning [12, 13].

Though sequential collaborative learning has such strengths, it still suffers from inference attacks where an adversary tries to obtain additional knowledge about the model or the data used for training [14]. Data confidentiality renders these attacks a breach of privacy [15]. Inference attacks can be performed in a *white-box* setting by *insiders*, e.g., participants in collaborative learning, or by *outsiders* in a *black-box* setting [16]. The white-box setting is more powerful since it possesses truly positive samples and can, therefore, develop a more accurate dataset to train the attack model [17].

However, mitigating or preventing inference attacks in collaborative learning is challenging and remains an open problem [6, 17]. Differential privacy (DP) is widely applied to protect record-level or participant-level privacy by adding a small noise to the training process or to the output result. The record-level privacy aims to hide the existence of each individual in a dataset, but deploying record-level DP can not protect property inference attacks. For participant-level privacy, the overall feature of a participant (institution) is protected, but participant-level DP requires thousands of participants to retain utility. For fraud detection, the number of parties is small (around 10), and *participant-level* DP results in low model utility. Instead, the secure multi-party computation technique is a candidate to hide the training order so that an adversary can not infer which party a record belongs to. It remains the utility of the model and protects the privacy of involved institutions, such as the general features of all clients in a specific bank. To determine their position in training, adversaries can derive knowledge utilizing the timing information, so-called side-channel timing attacks. Based on the time it takes to process

a given input and the resulting output, the adversary can infer how many training processes have been done to assume his position. To protect against such attacks, a random can be added to anonymize the training time.

The objective of this paper is to construct a framework and protocols to mitigate membership and property inference attacks and side-channel timing attacks in a small-scale sequential collaborative learning setting among n participants.

We assume a fixed group of $n+\ell$ parties are willing to collaboratively train a model of a predetermined type. Each party p_i owns a local dataset and has published its public key pk_{p_i}. We assume each party has sufficient computational power to train a model locally in a reasonable time, and the training time for each party is similar. All parties are subdivided into two groups: a group of ℓ leaders A_ℓ and a group of n participants p_i.

In the adversary model, we assume that the participants are *semi-honest* (honest-but-curious). All participants follow the protocol honestly but try to infer the knowledge of a target participant using information received during the protocol. We assume that i out of n participants (adversaries) collude with each other to exchange their knowledge. The adversaries try to infer their training position using timing-based side-channel attacks based on the training time of other participants. Other kinds of side-channel attacks are beyond the scope of this study. Also, the adversaries aim to use the updated model of each round of training to apply inference attacks to identify (1) which participant a record belongs to and (2) the property of a target participant. For the leaders, we assume that more than half of them follow the protocol correctly.

In this paper, we propose a novel framework and protocols to mitigate membership and property inference attacks and side-channel timing attacks in small-scale sequential collaborative learning among $n(\sim 10)$ participants using secure multi-party computation and differential privacy, as shown in Figure 5.1. A training order is determined jointly without revealing the final result to the participants. Together with the Tor anonymous communication network, sender anonymity is ensured, and an additional extension provides responder anonymity as well. As a result, the adversary is unable to target a specific participant. To jointly select a permutation, we leverage distributed decryption to ensure a ciphertext can only be decrypted if all parties contribute, and joint random number generation to mask individual values. Meanwhile, we apply differential privacy to add a random time delay of each training procedure to protect against the timing side channel. During the training procedure, record-level local differential privacy is utilized to double safeguard against membership inference attacks besides secure multi-party selection.

Our detailed security and privacy analysis illustrate that our design can mitigate both membership and property inference attacks, even when participants collude among a small number of parties. With only sender anonymity, adversaries have low confidence in their attacks since the training order is hidden. For example, when 2 out of 6 parties collude, we lower the success rate of membership inference attacks from 78% to 26% and from 24% to 12% concerning different training orders and targets. When i out of n parties collude, we illustrate and prove that the attack success rate is lowered in most scenarios and remains the same in the rest. Meanwhile, when responder anonymity is guaranteed, though the communication cost is roughly doubled, the attack success rate is reduced by a factor of α compared to scenarios where the training order is exposed. α

ranges from 2 to 4 and is dependent on the distribution of adversaries within the network topology. For property inference attacks, the attack success rate also decreases, with the effectiveness of our design influenced by the data distributions among participants. Our example shows an expected 30% lower attack performance.

In addition, our complexity analysis and experiments show that our protocol is practical and requires an overhead of only a few seconds considering nine participants. The overhead of anonymous communication ranges from seconds to a few minutes based on the model size (3s for 55 KB and 90s for 60 MB). To our knowledge, this work is the first to leverage secure multi-party computation with minimal overhead to provide privacy and robustness against inference attacks in sequential collaborative learning.

Remark 5.1. Even though we present our protocol in a fraud detection setting, it is versatile and can be used in any setting that requires sequential collaborative learning, such as in the medical and healthcare domain [4, 18].

Remark 5.2. In this paper, our protocols focus on *participant-level* privacy for a small number of institutions while *participant-level* differential privacy requires thousands of parties for a good utility [19]. Our protocols emphasize that an adversary can not infer which specific participant a record belongs to, but it is possible to execute inference attacks to find whether a record is in the training sets of all other $n-1$ parties. *Record-level* differential privacy is applied to mitigate this issue.

The remainder of this paper is structured as follows. We first introduce related works and preliminaries in Sections 5.2 and 5.3. Then, we explain our protocols and collaborative learning procedures in Section 5.4 and 5.5. Afterwards, we address security and privacy analysis in Section 5.6. Finally, we analyze complexity and performance in Sections 5.7 and 5.8, and conclude the paper in Section 5.9.

5.2. RELATED WORK

Inference Attacks. Inference attacks aim to infer knowledge about the data used to train a model or the model itself [19]. Two types of inference attacks can be distinguished: *membership inference attacks*, aiming to determine whether a specific entry is included in the training set, and *property inference attacks*, aiming to infer additional information that only holds for a subset of the training set and is not included as a feature [19]. To perform a membership inference attack on a *target model*, two models need to be trained, the *shadow model* and the *attack model* [15]. The shadow model is trained on data similar to the data used to train the target model. This ensures that predictions made by the shadow model are similar to those of the target model. To train the attack model, we use predictions obtained by the shadow model and provide these with a ground truth label, indicating whether the used sample was originally included in the training set. Since we have no knowledge of which samples were used to train the target model, a correctly trained shadow model is essential to provide ground truth labels for the attack model [15]. Similarly, for property inference attacks, attackers require data labelled according to the property they want to infer [19]. The vulnerability of parallel collaborative learning to insider membership and property inference attacks has been noted in several works [17, 19]. For sequential settings, Pustozerova and Mayer consider membership inference attacks [6]. The attack success is measured with varying numbers of participants,

epochs, and classes. These settings are compared to a baseline centralized setting to determine their influence on attack success. Results show that the attacks become more powerful when executed in a decentralized setting. Moreover, the attack performance increases as the number of epochs and the number of participants increase.

Mitigation Techniques. While we demonstrate the severe consequences of these attacks, mitigating or preventing them remains an open problem [6, 17]. Truex et al. suggest *model hardening techniques*, such as model choice and regularization, but do not evaluate them for effectiveness [17]. Melis et al. propose four different techniques to mitigate vulnerability [19], including sharing fewer gradients, dimensionality reduction, dropout, and differential privacy. The first three approaches either do not restrict attack success or significantly decrease model performance. With differential privacy, the first way is to deploy at *record-level* using noise to obfuscate individual records. Since the noise added is identically distributed for all dataset entries, this does not prevent property inference. To achieve this, *participant-level* differential privacy can be used. While record-level differential privacy ensures that the behaviour of a model does not change when one entry in the dataset is removed, participant-level differential privacy ensures model behaviour does not change when all data entries belonging to a specific user are removed [20]. From this, it follows that an attacker is unable to determine which data point belongs to which user. Therefore, by definition, participant-level differential privacy prevents property inference attacks. Unfortunately, to ensure model performance and participant-level differential privacy at the same time, thousands of participants are required [19]. In this work, we focus on a collaborative learning setting where the number of participants is in the order of tens, and the amount of noise required to ensure participant-level differential privacy is of such magnitude that it renders the model unusable. As noted in [21], more research is needed to prevent property inference attacks in collaborative settings with low numbers of participants. The work of [6] proposes to randomize the order of the nodes in each training cycle to reduce the information available to the attacker. The attacker does not know at which time a specific participant trains the model, thereby disabling them from inferring whether a data sample was held by that participant. While they reason this prevents membership inference attacks, they leave its evaluation to future work.

5.3. PRELIMINARIES

In this section, we introduce the cryptographic building blocks of the paper with notations in Table 5.1.

5.3.1. JOINT RANDOM NUMBER GENERATION

We rely on joint random number generation to mask individual values generated by participants. Based on the works of [22] and [23], we ensure these random numbers cancel out during aggregation, thereby allowing the aggregating party to obtain a correct result. The scheme uses the sign function:

$$s_i(j) = \begin{cases} 1 & \text{if } i > j\,, \\ -1 & \text{if } i < j\,. \end{cases} \tag{5.1}$$

Table 5.1: Notation table.

Symbol	Definition
p	large prime
g	generator for $\mathbb{Z}_p^*$
q	prime order of $\mathbb{Z}_p^*$
G	cyclic group with prime order q
ℓ	number of leaders
$A_1, \ldots, A_\ell$	group of leaders
n	number of participants
$p_1, \ldots, p_n$	group of participants
$n!$	number of permutations among participants
v_i	random number generated by p_i, $0 \leq v_i \leq (n!-1)$
v	jointly selected permutation index
π	permutation table containing all permutations
π_v	jointly selected permutation v from π
sk_{p_i} / sk_{A_i}	secret key held by p_i / A_i
pk_{p_i} / pk_{A_i}	public key held by p_i / A_i
PK	shared decryption key held by leaders
k_{p_i}	secret value generated by p_i
R_{p_i}	jointly generated random number of p_i
r_i	fresh randomness in $\mathbb{Z}_q^*$
$E(c_{1_{p_i}}, c_{2_{p_i}})$	ciphertext created by p_i
$H(c_1^{sk_{A_\ell}})$	commitment on c_1 by A_ℓ
$s_i(j)$	sign function (see Equation 5.1)

Joint random number generation proceeds as follows. First, each party broadcasts its public key $g^{k_{p_i}}$, for public generator $g \in \mathbb{Z}_p^*$ and a random secret key $k_{p_i} \in \mathbb{Z}_p^*$. Then, each party calculates its random number R_{p_i} using $s_i(j)$ that

$$R_{p_i} = \sum_{j=1, j\neq i}^{K} s_i(j)(g^{k_{p_j}})^{k_{p_i}} = \sum_{j=1, j\neq i}^{K} s_i(j) g^{k_{p_j} k_{p_i}} \tag{5.2}$$

where $g^{k_{p_j}}$ is the broadcast key from p_j, and K is the number of parties for the joint random number generation. Verifying resulting random numbers sum to zero is done by

$$\begin{aligned} \sum_{i=1}^{K} R_{p_i} &= \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} s_i(j) g^{k_{p_i} k_{p_j}} \\ &= \sum_{i=1}^{K} \sum_{j=1}^{i-1} (s_i(j) + s_j(i)) g^{k_{p_i} k_{p_j}} = 0\,. \end{aligned} \tag{5.3}$$

5.3.2. DISTRIBUTED EXPONENTIAL ELGAMAL CRYPTOSYSTEM

To allow the use of additive homomorphism, we use a distributed exponential version of the ElGamal cryptosystem [24]. This requires a cyclic group G with prime order q and public generator g to be agreed on by all participants.

Key Generation. Each party generates its secret key $sk_{p_i} = x_i \leftarrow [0,\dots,q-1]$ and its public key $pk_{p_i} = g^{x_i} \pmod p$. Then, a shared public key PK is created to be used for encryption and collaborative decryption.

$$PK = \prod_{i=1}^{n} pk_{p_i} = g^{x_1+\cdots+x_n} . \tag{5.4}$$

Encryption. In the exponential version of the ElGamal cryptosystem, a message $m \in G$ is encrypted as g^m . First, fresh randomness $r \in \mathbb{Z}_q^*$ is generated. Then, c_1 is computed as $c_1 = g^r$ and c_2 is computed as $c_2 = g^m \cdot PK^r$. Finally, the resulting ciphertext is $E(m) = (c_1, c_2)$.

Shared Decryption. To perform decryption, each party broadcasts a partial decryption of the ciphertext. These are combined to retrieve the correct plaintext message. For a ciphertext $E(m) = (c_1, c_2)$, this results in the following steps:

- Each party computes ${c_1}^{sk_{p_i}}$ and broadcasts this with commitment $H({c_1}^{sk_{p_i}})$.
- Each party checks whether all received partial decryptions ${c_1}^{sk_{p_i}}$ and corresponding commitments $H({c_1}^{sk_{p_i}})$ match.
- If this is the case, each party computes $\frac{c_2}{\prod_{i=1}^{n} {c_1}^{sk_{p_i}}} = \frac{c_2}{{c_1}^{sk_{p_1}+\cdots+sk_{p_n}}} = g^m$.
- To retrieve m, each party computes the discrete log.

Homomorphism. Exponential ElGamal is an additively homomorphic cryptosystem that multiplying two ciphertexts results in the addition of the two plaintexts: $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$, which can be verified as follows:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= (g^{r_1}, g^{m_1} \cdot PK^{r_1}) \cdot (g^{r_2}, g^{m_2} \cdot PK^{r_2}) \\ &= (g^{r_1+r_2}, g^{m_1+m_2} \cdot PK^{r_1+r_2}) = E(m_1 + m_2) . \end{aligned} \tag{5.5}$$

Additionally, additive homomorphism ensures that exponentiation of a ciphertext with a plaintext message corresponds to a multiplication of the two in the plaintext domain: $E(m_1)^{m_2} = E(m_1 \cdot m_2)$, which can be verified as follows:

$$\begin{aligned} E(m_1)^{m_2} &= (g^{r_1}, g^{m_1} \cdot PK^{r_1})^{m_2} \\ &= (g^{r_1 \cdot m_2}, g^{m_1 \cdot m_2} \cdot PK^{r_1 \cdot m_2}) = E(m_1 \cdot m_2) . \end{aligned} \tag{5.6}$$

5.3.3. MAJORITY VOTING

Each party uses majority voting to determine who to send their data to. In majority voting, parties vote once for one of the n candidates, and the party receiving the most votes

wins. Votes are represented by punch-hole vector ballots [25]. Each ballot consists of n options. When voting for candidate i, the option $C_i[i]$ is set to an encryption of one $C_i[1]$. All other options are set to $C_i[0]$. To allow additively homomorphic tallying of votes, exponential ElGamal encryption is used. To ensure a ballot is correctly formed (e.g. contains 1 one and $n-1$ zeros), non-interactive zero-knowledge (NIZK) proofs are included [24]. These allow a party to prove a ciphertext $E(m)$ is encrypted correctly using exponential ElGamal encryption. This means that a cyclic group G, with prime order q and public generator g, is available. Each party has a secret key x_i and public key $pk = g^{x_i}$. To prove a ciphertext $E(m) = (c_1, c_2) = (g^r, g^m \cdot pk^r)$ is encrypted from m, it is necessary to prove that c_1 and $\frac{c_2}{g^m}$ have the same exponent. This is done using the non-interactive zero-knowledge proof [24]. If verifications pass, $E(m) = (c_1, c_2)$ is a valid encryption of m.

NIZK proof for Plaintext Knowledge

Prover:

- Generate random $r \in \mathbb{Z}_q$
- Compute $E(m) = (c_1, c_2) = (g^r, g^m \cdot pk^r)$
- Generate random $t \in \mathbb{Z}_q$
- Compute $T_1 = g^t$ and $T_2 = pk^t$
- Compute $v = Hash(E(m)||T_1||T_2)$ and $s = r \cdot v + t$
- Send c_1, c_2, T_1, T_2, v and s to the Verifier

Verifier:

- Verify that $g^s = {c_1}^v \cdot T_1$
- Verify that $pk^s = (\frac{c_2}{g^m})^v \cdot T_2$

5.3.4. DIFFERENTIAL PRIVACY

Differential privacy [26, 27] aims to apply a small noise to a query to hide the existence of any single data in a dataset. For neighbouring datasets $\mathcal{D}$ and $\mathcal{D}$ which only differs in one record, a mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (DP) if and only if for any set $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}]. \tag{5.7}$$

The Laplace mechanism achieves ϵ-DP by applying the Laplacian noise to the output of a query.

Definition 5.1 (Laplace Mechanism [27]). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-DP over a query $f : \mathcal{D}^N \rightarrow \mathbb{R}^k$ if $y_i \sim \text{Lap}\left(\frac{\Delta f}{\epsilon}\right)$ and*

$$\mathcal{M}(\mathcal{D}, f, \epsilon) = f(\mathcal{D}) + (y_1 \ \ldots \ y_k). \tag{5.8}$$

Here $Lap(b)$ is the density $Lap(x|\mu = 0, b)$. Δf is the l_1 sensitivity of the query f as

$$\Delta f = \max_{D,D' \in \mathcal{D}^N : |D-D'| \leq 1} |f(D) - f(D')|. \tag{5.9}$$

Besides ϵ-DP, a small error factor δ can be introduced to release the strong guarantee a bit and make it more practical. A mechanism $\mathcal{M}$ satisfies (ϵ, δ)-DP if and only if for any set $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta. \tag{5.10}$$

Similar to the Laplace mechanism, the Gaussian mechanism achieves (ϵ, δ)-DP by applying the Gaussian noise.

Definition 5.2 (Gaussian Mechanism [27]). *A randomized algorithm $\mathcal{M}$ satisfies (ϵ, δ)-DP over a query $f : \mathcal{D}^N \to \mathbb{R}^k$ if $y_i \sim \mathcal{N}\left(\mu = 0, \sigma\right)$, $\sigma^2 = 2\ln(1.25/\delta) \cdot (\Delta_2 f)^2/(\epsilon^2)$ and*

$$\mathcal{M}(\mathcal{D},\ f,\ \epsilon) = f(\mathcal{D}) + (y_1\ \ldots\ y_k). \tag{5.11}$$

Here $\Delta_2 f$ is the l_2 sensitivity of the query f as

$$\begin{aligned} &\Delta_2 f = \max_{D,D' \in \mathcal{D}^N : |D-D'| \leq 1} ||f(D) - f(D')||_2, \\ &\text{and } ||\mathbf{x}||_2 = \sqrt{\Sigma_i |x_i|^2}\,. \end{aligned} \tag{5.12}$$

With differential privacy, ϵ is the privacy parameter where a lower value of ϵ can result in a larger noise and a stronger privacy guarantee. In this paper, we apply the Laplace mechanism to introduce the time delay and the Gaussian mechanism for differentially private learning.

5.4. TRAINING ORDER SELECTION

As explained in Section 5.1, we consider a setting with multiple parties that collaboratively train a predetermined machine learning model in a sequential setting. With this protocol, we prevent inference attacks by jointly choosing the training order without informing all parties about the result. To achieve this, we divide parties into two groups: *leaders* and *participants*. At the start of the protocol, leaders agree on a permutation table π containing all $n!$ permutations of participants, representing all possible orders for training. To determine the selected permutation, each participant submits a random number $0 \leq v_i \leq (n! - 1)$. Since sharing the permutation with each participant directly still enables the execution of inference attacks, participants are only informed on which participant is next in order. This is done using majority voting: each leader informs each participant to whom to send their data by voting on the respective participant. Participants send their gradient to the one that has received the most votes. This ensures the resulting order is correct as long as the majority of the leaders are honest. Our protocol broadly works as follows:

1. *Initialization*: A set of leaders A_ℓ is selected randomly. Together, these agree on the permutation table π. Additionally, public parameters for the encryption scheme are set. Based on these, all parties generate their public and secret key. Finally, leaders jointly generate their shared decryption key.

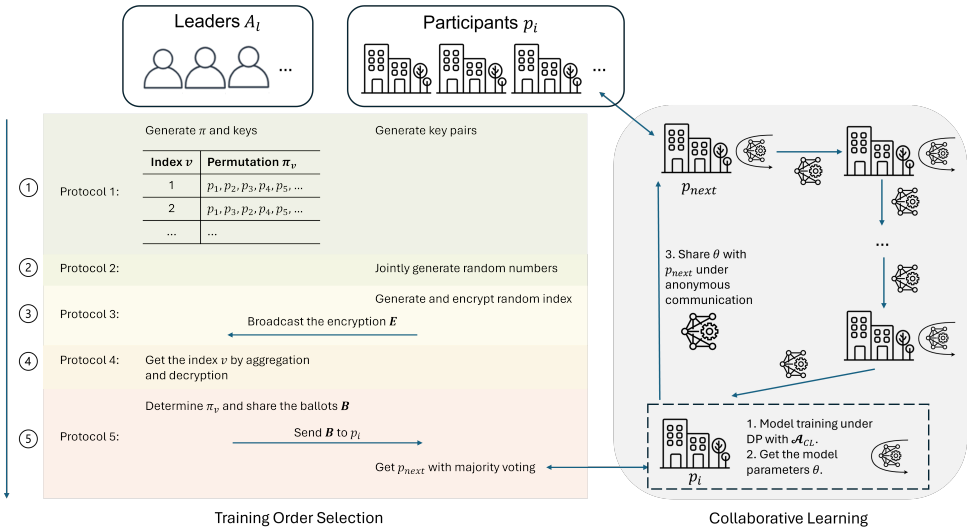


Figure 5.1: The framework for robust collaborative learning among a small number of participants with multi-party selection and differential privacy. The left part is the training order selection, which includes all five protocols. When the order is established, the right part is the collaborative learning procedure, including model training and anonymous communication.

2. *Joint Random Number Generation*: All participants jointly generate random numbers with a zero-sum. These are used to mask individual submissions.

3. *Joint Index Selection*: Each participant generates a random number $0 \leq v_i \leq (n! - 1)$ and encrypts it together with the jointly generated random number from the previous step. The resulting ciphertext is shared with the leaders.

4. *Aggregation & Partial Decryption*: Each leader uses additive homomorphism to sum received ciphertexts. Additionally, it shares a partial decryption of the result with all other leaders. Finally, each leader combines all received partial decryptions to determine v using the modulus of the sum.

5. *Majority Voting*: To inform each participant to whom to send their data, leaders send each participant a ballot. Each participant tallies the received votes and sends its data to the participant who received the most votes.

5.4.1. INITIALIZATION

First, the set of leaders $A_1, \ldots, A_\ell$ is selected randomly. Together, these agree on a permutation table π, containing all possible permutations $n!$, and on public parameters of the distributed exponential ElGamal cryptosystem. Next, all participants generate their secret and public key (sk_{p_i}, pk_{p_i}) using the key generation algorithm described in Section 5.3. Leaders generate their secret and public key (sk_{A_i}, pk_{A_i}) similarly. Finally, leaders generate their shared decryption key using Equation 5.4. An overview of this procedure is given in Protocol 4.

Protocol 4: Initialization

All leaders $A_1, \ldots, A_\ell$ do the following:

- Generate permutation table π
- Setup distributed exponential ElGamal (G, p, q, g)
- Generate public and secret key pair (sk_{A_i}, pk_{A_i})
- Generate shared decryption key $PK = \prod_{i=1}^{n} pk_{A_i} = g^{x_1+\cdots+x_n}$

All participants $p_1, \ldots, p_n$ do the following:

- Generate public and secret key pair (sk_{p_i}, pk_{p_i})

5.4.2. JOINT RANDOM NUMBER GENERATION

The joint random number generation (JRNG) step is performed among participants. Together, they derive random numbers that sum to zero. This allows the masking of individual values while ensuring their sum can still be derived. To achieve this, each participant first generates a secret value $k_{p_i} \in \mathbb{Z}_p^*$ and computes $g^{k_{p_i}}$. Next, each participant broadcasts $g^{k_{p_i}}$. Finally, each final random number R_{p_i} is computed using the sign function $s_i(j)$ and $R_{p_i} = \sum_{j=1, j\neq i}^{K} s_i(j) g^{k_{p_j} k_{p_i}}$ where K is the number of participants and k_{p_j} is the broadcast key from other participants. An overview of the JRNG procedure is given in Protocol 5.

Protocol 5: Joint Random Number Generation

All participants $p_1, \ldots, p_n$ do the following:

- Generate secret value k_{p_i} and broadcast $g^{k_{p_i}}$ to all participants
- Compute final random number using $R_{p_i} = \sum_{j=1, j\neq i}^{K} s_i(j) g^{k_{p_j} k_{p_i}}$

5.4.3. JOINT INDEX SELECTION

Each participant generates a random number $0 \leq v_i \leq (n! - 1)$, which is used later to determine the permutation π_v from the permutation table. First, this value is masked using the jointly generated random number R_{p_i}. Next, it is encrypted using the distributed exponential ElGamal cryptosystem and fresh randomness $r_i \in \mathbb{Z}_q^*$. Altogether, this yields $E(c_{1_{p_i}}, c_{2_{p_i}}) = (g^{r_i}, g^{v_i} \cdot PK^{r_i} \cdot g^{R_{p_i}})$, which is shared with all leaders. An overview of the procedure is in Protocol 6.

Protocol 6: Joint Index Selection

All participants $p_1, \ldots, p_n$ do the following:

- Generate random $0 \leq v_i \leq (n! - 1)$ and fresh randomness $r_i \in \mathbb{Z}_q^*$
- Encrypt $E(c_{1_{p_i}}, c_{2_{p_i}}) = (g^{r_i}, g^{v_i} \cdot PK^{r_i} \cdot g^{R_{p_i}})$
- Broadcast $E(c_{1_{p_i}}, c_{2_{p_i}})$

5.4.4. AGGREGATION & PARTIAL DECRYPTION

During this step, each leader aggregates all received ciphertexts using additive homomorphism. For aggregation, the masks computed using joint random number generation cancel out, allowing the correct addition of ciphertexts. This is verified as follows:

$$\begin{aligned} & E(m_{p_1}) \cdot \ldots \cdot E(m_{p_n}) \\ &= (g^{r_1}, g^{v_1} \cdot PK^{r_1} \cdot g^{R_{p_1}}) \cdot \ldots \cdot (g^{r_n}, g^{v_n} \cdot PK^{r_n} \cdot g^{R_{p_n}}) \\ &= (g^{r_1+\cdots+r_n}, g^{v_1+\cdots+v_n} \cdot PK^{r_1+\cdots+r_n} \cdot g^0) \\ &= E(m_{p_1} + \cdots + m_{p_n}) \,. \end{aligned} \tag{5.13}$$

After aggregation, each leader A_i publishes a partial decryption by broadcasting $c_1^{sk_{A_i}}$ with commitment $H(c_1^{sk_{A_i}})$. If all commitments are verified, decryption proceeds as follows:

$$\frac{c_2}{\prod_{i=1}^{\ell} c_1^{sk_{A_i}}} = \frac{c_2}{c_1^{sk_{A_1}+\cdots+sk_{A_n}}} = g^v \,. \tag{5.14}$$

Finally, v is determined using a lookup table. To ensure the sum of all numbers v_i, as submitted by participants, lies within the range $0 \leq v \leq (n! - 1)$, we use the modulus of the sum as $v = (\sum v_i) \pmod{n!}$ [28]. An overview of this procedure is shown in Protocol 7.

Protocol 7: Aggregation & Partial Decryption

All leaders $A_1, \ldots, A_\ell$ do the following:

- Aggregate all received ciphertexts $\prod_{i=1}^{n} E(m_{p_i})$
- Broadcast partial decryption and commitment $(c_1^{sk_{A_i}}, H(c_1^{sk_{A_i}}))$
- Computed complete decryption $\frac{c_2}{\prod_{i=1}^{\ell} c_1^{sk_{A_i}}} = g^v$
- Determine v using a lookup table

5.4.5. MAJORITY VOTING

Using the obtained index v, leaders determine the corresponding permutation π_v from π and use it to determine to which participant p_j a participant p_i should send its data. To inform participants about this, leaders create n punch-hole vector ballots B_{p_i} together with non-interactive zero-knowledge proofs.

After receiving ℓ ballots and verifying zero-knowledge proofs, participants perform homomorphic tallying of votes by forming matrices using received ballots. Each row represents a participant p_j, and each cell (p_j, A_i) indicates whether a leader A_i voted to send data to participant p_j or not. By summing each row using additive homomorphism, the number of leaders agreeing on a participant p_j is determined, informing p_i to whom to send its data. An overview of the majority voting step is given in Protocol 8.

Protocol 8: Majority Voting

All leaders $A_1, \ldots, A_\ell$ do the following:

- Determine π_v
- Create n ballots B_{p_i} and send these to the corresponding participant

All participants $p_1, \ldots, p_n$ do the following:

- Sum all received ballots B_{p_i} using additive homomorphism
- Send data to participant p_j with the most votes

5.4.6. AGAINST SIDE-CHANNEL TIMING ATTACKS

Even though the training order is not shared with participants, in practice, an adversary can infer his position using possible side channels such as the timing information. As shown in our adversary model in Section 5.1, we only consider the adversary trying to infer their position in the training order based on the training time of other participants. For example, an adversary aims to obtain a numeric code for which each number is between 0 and 9. Assume that the program used to check the passcode aborts immediately after receiving a wrong digit. The adversary derives the first digit by supplying ten codes, each with a different start number. The code with the most time to process is the one with the correct first digit. By repeating this process, the adversary retrieves the passcode using only the execution time of the given inputs.

Our protocol is vulnerable to side-channel timing attacks under the assumption that computations by each participant take similar time. After joint permutation selection, the first party starts training and sends gradients to the next participant. Each participant repeats this process following the determined permutation. If all participants own the same amount of data and use the same hardware for training, the training time is equal among the participants. Using this information, participants obtain their position in the permutation by dividing the total time they have waited by the time needed for training. Now, inference attacks can be executed by attacking the first and second models received and by comparing the results obtained in the attacks. For example, in a membership inference attack, participants p_2 and p_{n-1} can accurately attack participants p_1 and p_n. Additionally, all other participants can determine whether a data sample was held by the ones before or after them, thereby also acquiring secret knowledge.

To mitigate this vulnerability, we introduce random delays, which are determined using the Laplace mechanism $\text{Lap}(\mu, b = \Delta f/\epsilon)$ [29, 30]. By using a well selected value of ϵ, we provide differentially private time delays [27]. In general, the overall time t_i for a participant p_i includes the training time t_{tr_i} and the time delays t_{d_i}. With the Laplace noise $Y_i \sim \textbf{Lap}(\mu, b = \Delta f/\epsilon)$, we have:

$$t_i = t_{tr_i} + t_{d_i} = t_{tr_i} + Y_i. \tag{5.15}$$

If we assume that each participant has a similar training time of t_{avg} and the first participant is p_0 in the training order, participant p_i can infer its position i using its waiting time t_{w_i} and $i = t_{w_i}/t_{avg}$. With our random delays, $t_{w_i} = i \cdot (t_{avg} + Y_i)$ and

$$i = \frac{i \cdot (t_{avg} + Y_i)}{t_{avg}} = i + \frac{i \cdot Y_i}{t_{avg}}. \tag{5.16}$$

As a result, the inferred position error brought by the introduced random delay is $i \cdot Y_i / t_{avg}$. Here, the second participant has the shortest delay since $i = 1$. To hide the training order, Y_i / t_{avg} is supposed to be larger than k with a high probability. Note that if $k < 1$, the second participant can still infer its position in the training order, so $\Pr(|Y| > 1 \cdot t_{avg})$ needs to be large enough. Similarly, the delay is too long if k is larger than the number of participants n. We set $0 < k \leq n$ and choose the expected $k_e = \frac{n}{2}$. We have the noise Y_i that

$$Y_i \sim \textbf{Lap}(\mu = k_e \cdot t_{avg}, b = \Delta f / \epsilon)\ . \tag{5.17}$$

Note that we choose to use $\mu = k_e \cdot t_{avg}$ instead of 0 to avoid generating a delay shorter than t_{avg} (with $t < 1$). Also, we do not set $1 < k \leq n$ since the initial setting of k is public, and the second participant can still infer that it is in the second position if $1 < k < 2$. Instead, we choose to keep the probability of $k < 1$ to be low. With the probability density function of Laplace distribution, we have

$$\Pr(Y_i > k \cdot t_{avg}) = \frac{\int_{k \cdot t_{avg}}^{n \cdot t_{avg}} \frac{1}{2b} e^{-\frac{|y-\mu|}{b}} dy}{\int_{0}^{n \cdot t_{avg}} \frac{1}{2b} e^{-\frac{|y-\mu|}{b}} dy}\ , \tag{5.18}$$

and

$$F(y) = \int_{-\infty}^{y} \textbf{Lap}(\mu, b) = \begin{cases} \frac{1}{2} e^{-\frac{\mu - y}{b}} & \text{if } y < \mu\ , \\ 1 - \frac{1}{2} e^{-\frac{y-\mu}{b}} & \text{if } y \geq \mu\ . \end{cases} \tag{5.19}$$

With the assumption that all participants have similar training time, we have $t_{avg} \approx max\{t_{tr_i}\}$. Also, with $\mu = \frac{n}{2} t_{avg}, b = \frac{\Delta f}{\epsilon}$, we can get

$$\begin{aligned} \Pr(Y_i > k \cdot t_{avg}) &= \begin{cases} \frac{1 - \frac{1}{2} \cdot e^{-\frac{\mu}{b}} - \frac{1}{2} \cdot e^{-\frac{\mu - k t_{avg}}{b}}}{1 - e^{-\frac{\mu}{b}}} & \text{if } k t_{avg} < \mu\ , \\ \frac{\frac{1}{2} \cdot e^{-\frac{k t_{avg} - \mu}{b}} - \frac{1}{2} \cdot e^{-\frac{\mu}{b}}}{1 - e^{-\frac{\mu}{b}}} & \text{if } k t_{avg} \geq \mu\ , \end{cases} \\ &= \begin{cases} \frac{1 - \frac{1}{2} \cdot e^{-\frac{n\epsilon}{2}} - \frac{1}{2} \cdot e^{-(\frac{n}{2} - k)\epsilon}}{1 - e^{-\frac{n\epsilon}{2}}} & \text{if } k < \frac{n}{2}\ , \\ \frac{\frac{1}{2} \cdot e^{-(k - \frac{n}{2})\epsilon} - \frac{1}{2} \cdot e^{-\frac{n\epsilon}{2}}}{1 - e^{-\frac{n\epsilon}{2}}} & \text{if } k \geq \frac{n}{2}\ . \end{cases} \end{aligned} \tag{5.20}$$

Before the training starts, we can control the probability of outputting a delay longer than k average training time. Then, the value of ϵ can be calculated using Equation (5.20). With a large number of Pr and k, there is a low probability that the participant can distinguish between different participants based on the training time. We include further security and privacy analysis in Theorem 5.2 in Section 5.6.

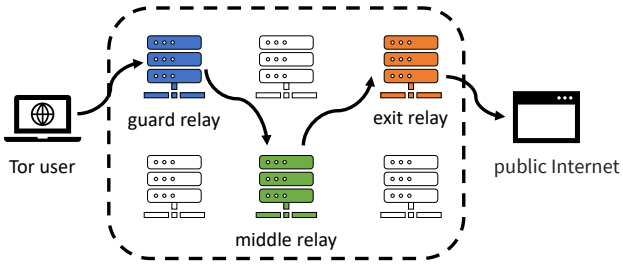


Figure 5.2: An example that a user browses a website over Tor where the traffic goes over multiple relays.

Meanwhile, a large time delay is needed only for the first one or two rounds of training to hide the order. When the first few rounds are finished, the value of Δf can be updated based on the $(max - min)$ time of previous rounds. To ensure each participant has access to the correct value Δf, we require leaders to share it with the participants.

5.5. COLLABORATIVE LEARNING

In Section 5.4, we introduce our privacy-preserving training order selection method. After the training order is determined, collaborative learning is carried out among a small number of parties. As shown in Figure 5.1, the participant p_i is supposed to train the model under differential privacy and share the updated parameters with the next participant p_{next} under anonymous communication. In this section, we further explain anonymous communication and training procedures.

5.5.1. ANONYMOUS COMMUNICATION

With our protocol, participants only receive information on the next receiver of the model. This implies they have no knowledge of whose data they are receiving. Otherwise, colluding participants can utilize the information to infer their position in the network and break the security guarantees in Section 5.4. As a result, it is important to provide *sender anonymity* using anonymous communication channels. Sender anonymity ensures that a message cannot be linked to its sender [31]. This is achieved by requiring participants to send their model updates over an anonymous network.

In this paper, we apply Tor [32] to achieve sender anonymity among the participants. Tor is a circuit-based, low-latency distributed overlay network for anonymous communication, and it currently stands as the most widely used system of its kind due to its low latency and flexibility [33]. In a typical Tor circuit, the traffic from a Tor user is sent through three randomly selected relays before reaching its destination, as illustrated in Figure 5.2. These relays include the guard relay, the middle relay, and the exit relay. Each relay only knows its predecessor and successor, making it impossible to trace the complete path from the user to the destination. The message is encrypted in multiple layers, with each layer encrypted with the shared symmetric key between the user and a specific relay. The user first selects a guard relay, establishes the connection, and performs a Diffie-Hellman key exchange to derive the shared symmetric key. For the middle and exit relays, the connection is established through the guard relay (and subsequently through

the middle relay). Each relay decrypts its respective layer to reveal the address of the next relay and forwards the remaining encrypted message. The exit relay can access the original traffic from the user and send it out onto the public Internet. During the transmission, the sender's IP is only known by the guard relay, but the guard relay does not know the content or the destination of the message since it can not decrypt the subsequent layers of encryption. As a result, sender anonymity is achieved.

Compared to Onion Routing (OR) [34], Tor applies the Diffie-Hellman key exchange instead of asymmetric encryption to achieve forward secrecy. Besides sender anonymity, Tor also provides hidden services to achieve responder anonymity, where the responder (hidden service host) can provide TCP service without revealing the IP address. The user and the responder can establish bidirectional anonymous communication through a rendezvous point, which is a randomly selected Tor relay. The responder randomly selects several introduction points and builds anonymous circuits to them. It then creates a document listing all its introduction points and uploads it to a set of HSDir nodes. The user can build an anonymous circuit to one of the introduction points and send a request that includes a randomly selected rendezvous point. The responder receives the request from the introduction point. Both the user and responder can build anonymous circuits to the rendezvous point and complete the handshake.

When the hidden service is not used, all participants can obtain the address of the next participant in the training order from the leaders, and they can establish anonymous communication with their next participants through Tor. The current participant encrypts the updated model using the receiver's public key to prevent the exit relay from accessing the plaintext of the model parameters. The encrypted model parameters are then sent to the next participant through Tor.

When the hidden service is used to achieve responder anonymity, the leaders send the encoded name of the next participant (hidden service host). The current participant encrypts the model parameters and sends them to the hidden service host. If the public key can potentially reveal the receiver's identity, a secure data communication protocol, such as HTTPS, is required to avoid the exit relay getting the model parameters. Although the hidden service provides responder anonymity, it also lowers the communication efficiency, as both the sender and the host need three relays in the anonymous circuit to reach the rendezvous point, so the communication cost is doubled.

Remark 5.3. Note that we show the feasibility of applying Tor in our framework, but optimizing Tor is not the focus of this paper. Meanwhile, other techniques, such as DC-net-based solutions [35], can also achieve sender anonymity and are applicable to our proposal. Nevertheless, Tor is already widely available, making it easier to integrate and generally more efficient [32, 35].

5.5.2. TRAINING PROCEDURE

In the collaborative learning procedure in Figure 5.1, the model is trained under differential privacy to protect against membership inference attacks. In the sequential collaborative learning setting, there is no central server to aggregate all updated model parameters, which makes it difficult to deploy central differential privacy. Instead, local differential privacy is considered during the training of each participant to address membership inference attacks.

In Algorithm 6, we illustrate the behaviour of a participant to achieve local differential privacy in collaborative learning based on DP-SGD [36]. The training parameter θ is initialized as a random value at the beginning of the collaborative learning. As shown in Figure 5.1, each participant receives θ from the previous one and uses it as the initial state for its training. The output θ is then encrypted and transmitted to the next receiver in the network. During the training process, The Gaussian mechanism is applied to deploy the noise to the gradient. Meanwhile, the gradient is bounded (with clipping) to avoid the excessive influence of a sample on the gradient. Since the training repeats E epochs, the composition of differential privacy is considered, and Algorithm 6 achieves $(O(p\sqrt{E}\epsilon), \delta)$-differential privacy [36].

Algorithm 6: $\mathcal{A}_{CL}$: Collaborative learning under local differential privacy with DP-SGD

Input: Dataset $\mathcal{D}: \{d_0, d_1, \ldots, d_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\Sigma_i \mathcal{L}(\theta, d_i)$, sampling probability p, learning rate η, Gaussian noise scale σ, clipping bound C, iterations E, training parameter θ_0.

Output: θ_E

for epoch $e \in [E]$ **do**

  for d_i in a random batch from $\mathcal{D}$ with probability p **do**

    $g_e(d_i) = \nabla_{\theta_e} \mathcal{L}(\theta_e, d_i)$

  end for

  $\tilde{g_e} = \frac{1}{pN} \sum_i (g_e(d_i) \min(1, \frac{C}{||g_e(d_i)||_2}) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

  $\theta_{e+1} = \theta_e - \eta \tilde{g_e}$

end for

return θ_E

Algorithm 6 provides a differentially private solution for deep learning under collaborative learning settings. Existing works show that the application of differential privacy can protect against membership inference attacks [37]. We give further privacy analysis in Section 5.6 on how it works under our framework and how privacy parameters should be selected.

Remark 5.4. Note that other machine learning models can also be considered here for collaborative learning, and differential privacy should be applied differently. This paper considers DP-SGD due to its good performance in different machine-learning tasks. Our security and privacy analysis is also based on the use of DP-SGD.

5.6. SECURITY AND PRIVACY ANALYSIS

In this section, we give a further analysis of our proposed protocols. We first show the security of our protocols in terms of the random permutation generation and selection. Besides, we further analyze the robustness of our protocols against side-channel timing attacks and inference attacks.

5.6.1. ROBUSTNESS AGAINST SIDE-CHANNEL TIMING ATTACKS

Lemma 5.1. *The order of permutation is randomly selected and cannot be accessed or changed by a probabilistic polynomial time (PPT) adversary $\mathcal{A}$. Each participant follows the correct order as long as the majority of leaders act correctly.*

Proof. The security of our protocol relies on two sub-protocols: the joint random number generation protocol (in Protocol 5) and the joint index selection protocol (in Protocol 6). During joint index selection, we assume each party uses a secure pseudo-random number generator, thereby ensuring the generated submissions are truly random. When doing so, the security of our protocol is reduced to that of the joint random number generation protocol, which is based on the hardness of the Decisional Diffie-Hellman Assumption. Since the jointly generated random numbers are determined using the public keys of the Diffie-Hellman Key Exchange, this ensures that the jointly generated numbers cannot be distinguished from random numbers. This shows that the permutation is chosen randomly in a semi-honest setting. Meanwhile, to ensure participants are correctly informed on whom to send their data to, the protocol relies on the security of the majority voting protocol. Based on the correctness of non-interactive zero-knowledge proofs, leaders prove that the ciphertext is indeed encrypted from its plaintext value, which ensures that each ballot is correctly formed and can not be manipulated by malicious leaders. With the majority voting, as long as the majority of leaders inform participants correctly, the outcome vote is correct. □

Lemma 5.2. *With properly selected parameters, the side-channel timing attack is mitigated under random delays.*

Proof. According to our adversary model in Section 5.1, the adversary uses the side-channel timing information, such as the training time of other participants, to infer its position in the training order. Equation (5.20) illustrates that the random delay introduces an error of at least k positions with probability $\Pr(Y_i > k \cdot t_{avg})$ when the adversary infers its position in the training order. However, if $k < 1$, the adversary can still infer that it is the second in the training order, so the probability $\Pr(Y_i < t_{avg})$ should be small enough. Here we set the probability larger than pr_k for outputting a delay more than $k \cdot t_{avg}$ ($k < n/2$), and we have

$$\begin{aligned} \Pr(Y_i > k \cdot t_{avg}) &= \frac{1 - \frac{1}{2} \cdot e^{-\frac{n\epsilon}{2}} - \frac{1}{2} \cdot e^{-(\frac{n}{2}-k)\epsilon}}{1 - e^{-\frac{n\epsilon}{2}}} > pr_k \\ \Rightarrow \frac{1}{2} n\epsilon &> \ln(\frac{1}{2} e^{k\epsilon} + \frac{1}{2} - pr_k) - \ln(1 - pr_k) \,. \end{aligned} \tag{5.21}$$

Equation (5.21) provides the approach to calculate the value of ϵ when there are n participants with any predetermined probability pr_k. Here we only care about the probability when k is small, such as $k < 1$, since $\Pr(Y_i < t_{avg})$ determines whether the second participant can infer its position in the order. In Tables 5.2 and 5.3, we show the examples where we limit the probability of $k < 1$ to 5% or 2%. The corresponding value of ϵ is 0.302 and 0.624, which provide considerable privacy guarantees.

Even if the adversary has access to the probability distribution as shown in Tables 5.2 and 5.3, they cannot confidently infer their position in the training order. Meanwhile, the

k	1	2	3	4	5	6	7	8	9	10
pr_k	5.0%	6.8%	9.1%	12.4%	16.7%	16.7%	12.4%	9.1%	6.8%	5.0%

Table 5.2: Example probability distribution with $pr_1 = 5\%$ and $\epsilon = 0.302$.

k	1	2	3	4	5	6	7	8	9	10
pr_k	2.0%	3.7%	7.0%	13.0%	24.3%	24.3%	13.0%	7.0%	3.7%	2.0%

Table 5.3: Example probability distribution with $pr_1 = 2\%$ and $\epsilon = 0.624$.

probability of pr_1 can be chosen based on different scenarios to minimize the potential privacy risk, which greatly mitigates the side-channel timing attack.

When the first one or two rounds are finished, the delay is much smaller since Δf is $max - min$ training time. Since the delay is differentially private and differs each round, a participant is unable to infer its value using information obtained over subsequent rounds. Altogether, this shows that the discussed side-channel timing attacks are mitigated using random time delays. □

Lemma 5.3. *All participants can only know the identity of the following receiver and can not infer the training order.*

Proof. Based on Lemma 5.1 and Lemma 5.2, the training order is hidden in the training order selection procedure. Also, the side-channel timing attack is mitigated during collaborative learning under anonymous communication networks. As a result, each participant can only get the identity of the next receiver in the network and can not infer the training order. □

5.6.2. ROBUSTNESS AGAINST MEMBERSHIP INFERENCE ATTACKS

With Lemma 5.1 to 5.3, we can further show the robustness of our protocols against inference attacks. Membership inference attacks aim to gain additional information about individuals from the output of the model, such as whether an individual is included in the training set of a participant. In sequential collaborative learning, each participant receives the model at different stages. According to Lemma 5.1 and Lemma 5.2, the training order is randomly selected and is unknown to all participants. As a result, each participant can only access the difference before and after one round is finished. Even though the adversary can extract information about records from the model, it is not possible to distinguish which specific participant the record belongs to. We examine three scenarios to demonstrate robustness against inference attacks. For scenarios involving colluding participants, we initially focus on sender anonymity in anonymous communication, and then expand the discussion to include responder anonymity.

NO PARTICIPANTS COLLUDE

When no participants collude in the network, a malicious participant $\mathcal{A}$ can only infer whether an individual is included in the training set of all other parties. Suppose there are n participants in the network, and the precision for membership inference attack is $a\%$ to correctly identify whether an individual is included in the aggregated dataset of

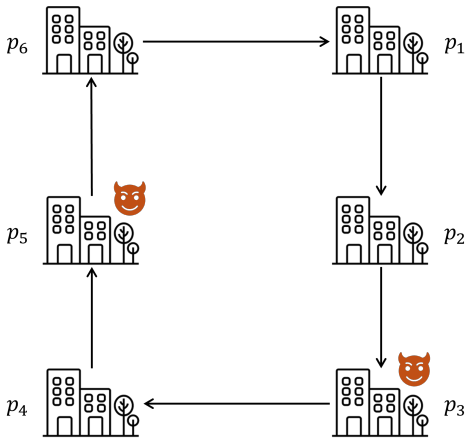


Figure 5.3: An example of collaborative learning with six participants where two of them (p_3, p_5) are colluded.

other $n-1$ participants. In this case, the adversary $\mathcal{A}$ can a% correctly identify whether an individual is included in the training set of other $n-1$ participants. Our protocols protect certain membership inferences based on our differential privacy-based learning algorithm. Meanwhile, $\mathcal{A}$ can $\frac{a\%}{n-1}$ correctly identify whether an individual is included in the training set of a specific participant since the training order is hidden and $\mathcal{A}$ can only access the aggregated updated model from all other participants. Thus, $\mathcal{A}$ can only randomly guess which participant the data belongs to. Without our protocols, $\mathcal{A}$ can get the training order of the sequential collaborative learning. In the first round, $\mathcal{A}$ can infer the previous participants with higher probability. More specifically, if $\mathcal{A}$ is the second in the order, he can a% correctly infer whether an individual is included in the training set of the first participant.

TWO PARTICIPANTS COLLUDE

In Figure 5.3, we give an example where two out of six participants collude and try to carry out membership inference attacks over other participants. With our protocols, if the adversaries $\mathcal{A}$ (p_3 and p_5) want to infer whether an individual is included in the training data of p_4 (or p_6), they do not know whether p_4 is the only participant between them. As a result, $\mathcal{A}$ can only achieve an attack precision guarantee of $\frac{a\%}{6-2-1} = \frac{a\%}{3}$ to correctly identify whether the individual is trained with p_4. Here, 6 is the number of all participants, 2 is the number of adversaries, and 1 is the number of participants (p_6) which is definitely not between the adversaries. Similarly, if $\mathcal{A}$ wants to infer p_1 (or p_2), the attack precision guarantee is $\frac{1}{2} \cdot \frac{a\%}{6-2-1} = \frac{a\%}{6}$ since $\mathcal{A}$ do not know whether p_1 is in the interval from p_5 to p_3 or in the interval from p_3 to p_5. Without our protocols, the adversary can access the training order and can a% correctly infer p_4. For p_1, p_2, p_6, the precision is $\frac{a\%}{3}$. Furthermore, with the responder anonymity property provided by Tor, the adversaries do not know the identity of the receiver for their updated model, and the precision of the inference attack is $\frac{a\%}{6-2} = \frac{a\%}{4}$.

Note that the estimation does not consider the influence of the size of the dataset on

the precision of membership inference attacks. With a CNN model trained on the CIFAR-10 dataset, the attack precision decreases when the training set size becomes larger [15]. Assume each participant holds around 2,000 data records. The attack precision for the aggregated model of p_1, p_2, p_6 is around 73% while the precision of p_4 alone is around 78% [15]. As a result, our estimation using $a\%$ is approximate instead of precise. Meanwhile, for p_1, we can roughly lower the precision from 73%/3 = 24.3% to 73%/6 = 12.2% and for p_4, the precision lowers from 78% to 78%/3 = 26%.

i OUT OF n PARTICIPANTS COLLUDE

Here we consider i colluded participants are not next to each other. Otherwise, we can see them as the same malicious participant. If the adversaries $\mathcal{A}$ want to infer a participant p_j who is not in their sending list, it is unclear between which two adversaries the target p_j is. There are i possible intervals (between two adversary participants) where p_j can be located. The attack precision guarantee is $\frac{1}{i} \cdot \frac{a\%}{n-2i+1}$ since each interval can be up to $n-2i+1$ participants. If p_j is in the sending list of one of the adversaries, the adversary can only ignore the other participants on the sending list. As a result, the precision guarantee is $\frac{a\%}{n-2i+1}$. Without our protocols, the adversary can access the training order, and the precision is based on the distribution of the adversaries. Based on random order generation, the expected precision is $E(a) = \sum_{x=1}^{n-2i+1} \frac{C_{n-2i}^{x-1}}{2^{n-2i}} \cdot \frac{a\%}{x}$. Here, C_n^i is the number of ways to choose i items from a set of n. $\frac{C_{n-2i}^{x-1}}{2^{n-2i}}$ is the probability that the target is in an interval with x participants. $\frac{a\%}{x}$ is the attack precision for this interval.

Note that we can also calculate the expected attack precision when our protocols are applied, which is $E(a)$ for participants on the sending list and $\frac{1}{i}E(a)$ for those not on the sending list of the adversaries. However, this can not represent the actual attack precision rate with our protocols since the adversaries can never know the actual number of participants between them. When the adversaries aim to carry out membership inference attacks on a specific target, the attack precision is not converged to $E(a)$ since the training order and topology are never changed, so we use $\frac{1}{i} \cdot \frac{a\%}{n-i}$ and $\frac{a\%}{n-2i+1}$ to represent the attack precision guarantees for participants in or not in the sending list. Compared to $E(a)$, we achieve equal or lower expected attack precision and the actual attack precision guarantees are even lower.

Similarly, with the responder anonymity, the adversaries can only infer that the target is among the other $n-i$ participants, so the precision of the inference attack is $\frac{a\%}{n-i}$. Compared to the expected precision without our protocols, we have the attack precision reduced by a factor of $\alpha = E(a)/\frac{a\%}{n-i}$ with examples in Table 5.4.

	$n=7$	$n=8$	$n=9$	$n=10$
$i=2$	2.34	2.33	2.30	2.27
$i=3$	3.00	2.92	2.81	2.71
$i=4$	3.00*	4.00	3.75	3.50

Table 5.4: Example ratios α for $n \in \{7,8,9,10\}$ and $i \in \{2,3,4\}$. For $n=7, i=4$, there are at least two neighbouring adversaries. This is equivalent to the $n=6, i=3$ case.

Our analysis of the three different scenarios gives both theoretical and experimental

results to demonstrate that our proposed protocol and framework mitigate membership inference attacks under sequential collaborative learning settings.

Remark 5.5. In Section 5.5, we apply local differential privacy to protect against membership inference attacks to infer whether an individual is included in the training set of all other participants. Based on our privacy analysis, we illustrate that we provide a double safeguard using our proposed multi-party selection protocols to further protect against membership inference attacks aiming at a specific participant.

5.6.3. ROBUSTNESS AGAINST PROPERTY INFERENCE ATTACKS

Different from membership inference attacks, property inference attacks aim to infer the property of a participant in the network, such as finding the ratio of characteristics (such as gender labels) in the dataset of a participant. Our protocols also provide robustness against property inference attacks.

With the example in Figure 5.3, p_3 and p_5 are malicious, and they want to infer the property of other participants. With our protocol, the training order is hidden from the adversaries. Even though the adversaries can infer the property between them, they do not know whether it is the property of the dataset of the target participant or it is the property of an aggregated dataset from multiple participants. For example, assume the sizes of datasets for p_1, p_2, p_4, p_6 are the same, and the female proportion of the datasets are $60\%, 85\%, 30\%, 50\%$ respectively. When the adversaries carry out the property inference attack to infer the property of p_1, they can only achieve an inference based on an aggregated dataset with female proportion 55%, which is far from the actual, which is 85%. As a result, the inference is not credible any more. Similarly, when the adversaries want to infer the property of p_4, though the proportion of the dataset is the same as the actual, the adversaries do not hold this information and can not have a confident inference as well.

Without our protocol, the adversaries can access the training order and find out which participants are between the adversaries. With the same example in Figure 5.3, the adversaries can correctly infer the property of p_4. Meanwhile, for the property inference of p_1, the adversaries can also improve their inference based on possible background knowledge on the distribution of datasets of p_2 and p_6.

As a result, our protocols hide the information about the training order to confuse the adversaries so that they can only guess the target positions with low confidence.

5.7. COMPLEXITY ANALYSIS

We analyze the computation and communication complexity of our protocol. For computation complexity, we use the number of modular operations and encryption and decryption operations. For communication complexity, we use the number of messages sent. In our analysis, we distinguish between operations performed by leaders and operations performed by participants. An overview of the computation and communication complexity of our protocol is given in Table 5.5.

For leaders, computation complexity is dominated by the number of modular exponentiation operations. For participants, it is dominated by the number of modular additions. Since $\ell \leq n$, computation complexity is higher for participants. Communication

Table 5.5: Computation and communication complexity of our protocol.

	Modular Addition	Modular Multiplication	Modular Exponentiation	Encryption/ Decryption	Number of Messages
Leaders $A_1,\ldots,A_\ell$					
1. Initialization	-	ℓ	1	-	$\ell-1$
2. Joint Random Number Generation	-	-	-	-	-
3. Joint Index Selection	-	-	-	-	-
4. Aggregation & Partial Decryption	-	n	1	-/1	$\ell-1$
5. Majority Voting	-	-	-	$n\cdot(n-1)$/-	n
Participants $p_1,\ldots,p_n$					
1. Initialization	-	-	1	-	-
2. Joint Random Number Generation	$n-1$	-	1	-	$n-1$
3. Joint Index Selection	-	-	-	1/-	ℓ
4. Aggregation & Partial Decryption	-	-	-	-	-
5. Majority Voting	-	$\ell\cdot(n-1)$	-	-/$\ell\cdot(n-1)$	1

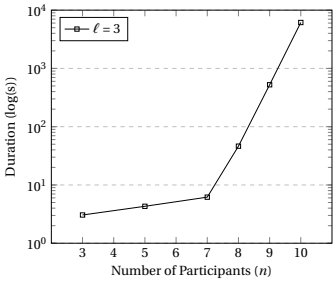
complexity is the highest for leaders since they need to include non-interactive zero-knowledge proofs. For example, to prevent one leader from corrupting the outcome of the protocol, we ensure ciphertexts can only be decrypted using partial decryption results from all other leaders. While this does ensure a correct outcome, it also requires ℓ additional messages to be sent during the aggregation & partial decryption protocol. Moreover, the requirement of shared decryption also causes high communication complexity during the initialization protocol: to establish a shared decryption key ℓ messages are shared among leaders.

5.8. PERFORMANCE ANALYSIS

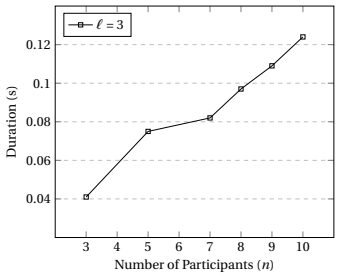
We implement our protocols and measure the run time of each step for varying numbers of participants and leaders. The number of leaders evaluated is $\{2,3,5,7,8,9,10\}$, while the number of participants varies between $\{3,5,7,8,9,10\}$. For each experiment, the number of leaders is less than or equal to the number of participants, $\ell \leq n$. Each setting was run ten times to prevent the influence of outliers. Manual inspection of the results showed the run time of the protocol is independent of the number of leaders ℓ. Therefore, in this analysis, we fix the number of leaders to $\ell = 3$. Experiments are conducted using a 2048-bit key. We test on a Dell XPS 15 9550 laptop with an Intel Core i7 CPU. We implement our protocols using Python 3.9 and the gmpy2 library[1] for modular arithmetic.

Initialization. The run time of initialization is dominated by pre-computations, such as the generation of the lookup table. Its length is defined as $n \cdot n!$, where n denotes the number of participants. As the number of participants increases, the length of the table increases with a factorial factor. Because of this, the time needed for initialization increases similarly. Figure 5.4a indeed shows that when increasing the number of participants from $n = 7$ to $n = 8$, the average duration of the initialization protocol increases significantly. In our design, we focus on small-scale collaborative learning where $n < 10$. Also, the initialization is only needed once when the training starts, and it is not needed for each round of training.

[1] https://gmpy2.readthedocs.io/en/latest/

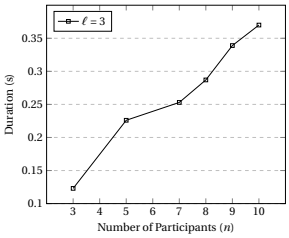


(a) Average duration (log(s)) of initialization.

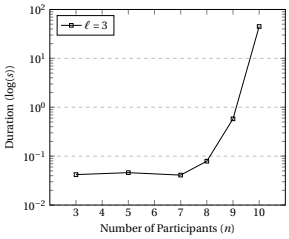


(b) Average duration (s) of joint random number generation.

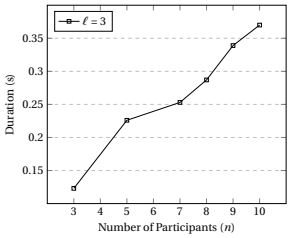
Figure 5.4: Average duration of initialization and joint random number generation protocols over 10 runs for varying numbers of participants (n).



(a) Average duration (s) of joint index selection.



(b) Average duration (log(s)) of aggregation & partial decryption.



(c) Average duration (s) of majority voting.

Figure 5.5: Average duration of joint index selection, aggregation & partial decryption and majority voting protocols over 10 runs for varying numbers of participants (n).

Joint Random Number Generation. Since joint random number generation is performed among participants only, its duration increases as the number of participants increases. However, since the number of operations does not increase with a factorial factor, we expect its runtime to increase slower. This is also visible from Figure 5.4b. Overall the protocol requires less than half a second to finish.

Joint Index Selection. The heaviest computation in this protocol is encryption, performed by participants using their generated numbers v_i. After encryption, each participant shares its ciphertext with all leaders. The average duration of the joint index selection protocol is shown in Figure 5.5a. Its duration increases as the number of participants increases. The required encryption operations also explain why this protocol requires more run time compared to joint random number generation.

Aggregation & Partial Decryption. During aggregation & partial decryption, each leader first performs partial decryption and full decryption afterwards. Finally, to determine the value for v, each leader uses the lookup table. From this, it follows that its duration grows significantly as the number of participants increases. This is also visible from Figure 5.5b.

Since we perform a lookup rather than generate the table itself, the runtime duration is less than initialization.

Majority Voting. At the start of the majority voting protocol, each leader creates n encrypted ballots and shares them with corresponding participants. After receiving ℓ ballots, each participant sums these and decrypts the result. As the number of participants increases, the number of ballots increases with it, resulting in more encryption operations being performed. Figure 5.5c shows that the duration of the majority voting protocol is similar to that of joint index selection.

Anonymous Communication. The use of Tor brings an overhead of approximately 3 seconds when accessing a 55-kilobyte website compared to a direct visit [32]. The overhead increases to around 90 seconds (from 210s to 300s) when downloading a file with 60 megabytes [32]. When the hidden service is used, we expect the overhead to be roughly doubled, as six relays instead of three are required.

Overall Performance. So far, we assessed each protocol separately for varying numbers of leaders and participants. While the number of leaders varied starting from two, the number of participants varied from three. This is required because in a setting with two participants, inferring whether a sample is owned by the other participant is obvious. Looking at each protocol separately shows that all protocols run in at most half a second, except for the initialization and aggregation & partial decryption protocols. It is important to note that initialization has to be performed only once. Additionally, the duration of aggregation & partial decryption requires less than a second for all $n \leq 9$. Overall, after initialization, our cryptographic protocols introduce an overhead of only a few seconds, while anonymous communication brings an overhead ranging from several seconds to a few minutes, depending on the model size.

5.9. CONCLUSION AND DISCUSSION

The performance of automated fraud detection is hindered by a lack of positive samples in collected data. While sequential collaborative learning can address this issue, its vulnerability to inference attacks restricts its applicability. In this paper, we propose a framework and protocols to mitigate inference attacks using secure multi-party computation, anonymous communication, and differential privacy. Our protocols require parties to jointly determine the training order while ensuring that participants only know the recipient of their model updates. Tor is applied to achieve sender anonymity during the transmission of model parameters. With its hidden service, responder anonymity is also guaranteed. Additionally, differential privacy is employed to mitigate side-channel timing attacks. Altogether, our design prevents participants from identifying the origin of model parameters or using updates from a specific target for model training, thereby mitigating inference attacks. Our detailed security and privacy analysis demonstrate that our design effectively mitigates inference attacks and side-channel timing attacks among semi-honest participants and honest majority leaders. Also, our experimental results illustrate that our protocol is practical and facilitates privacy-preserving sequential collaborative learning with an overhead of only a few seconds for our protocols and an additional overhead of several seconds for anonymous communication. To our knowl-

edge, our work is the first to prevent inference attacks using a secure joint permutation selection protocol.

In general, membership inference attacks aim to find whether an individual is included in the training set of a model. In this paper, we consider a slightly different membership inference attack that tries to find to which specific participant the record belongs. In practice, such inference is more useful and sensitive since it can link the information of an individual to a participant (such as companies, banks, or hospitals). In collaborative learning settings, it is more important to hide the identity of the participant who owns the data since this may expose sensitive information about the relationship between individuals and institutions. In real life, different companies, banks, and hospitals need small-scale collaborative learning for cooperative research or data analysis, which also highlights the need for our framework.

REFERENCES

[1] E. C. Bank. *Sixth report on card fraud*. 2020.

[2] H. Abouloifa and M. Bahaj. "Fraud Detection in Supply Chain 4.0: A Machine Learning Model". In: *International Conference on Advanced Intelligent Systems for Sustainable Development*. Cham: Springer Nature Switzerland, 2023, pp. 200–206. ISBN: 978-3-031-35245-4.

[3] A. D. Pozzolo, O. Caelen, Y. L. Borgne, S. Waterschoot, and G. Bontempi. "Learned lessons in credit card fraud detection from a practitioner perspective". In: *Expert Syst. Appl.* 41.10 (2014), pp. 4915–4928.

[4] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, et al. "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data". In: *Scientific reports* 10.1 (2020), pp. 1–12.

[5] European Parliament and Council of European Union. *Regulation (EU) 2016/679*. 2016.

[6] A. Pustozerova and R. Mayer. "Information leaks in federated learning". In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2020*. Vol. 10. 2020, p. 122.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*. Vol. 54. 2017, pp. 1273–1282.

[8] Y. Li and X. Lyu. "Convergence Analysis of Sequential Federated Learning on Heterogeneous Data". In: *Advances in Neural Information Processing Systems 36, NeurIPS 2023*. 2023.

[9] J. Lee, J. Oh, S. Lim, S. Yun, and J. Lee. "TornadoAggregate: Accurate and Scalable Federated Learning via the Ring-Based Architecture". In: *CoRR* abs/2012.03214 (2020). arXiv: 2012.03214.

[10] R. Zaccone, A. Rizzardi, D. Caldarola, M. Ciccone, and B. Caputo. "Speeding up Heterogeneous Federated Learning with Sequentially Trained Superclients". In: *26th International Conference on Pattern Recognition, ICPR 2022*. IEEE, 2022, pp. 3376–3382.

[11] M. Kamp, J. Fischer, and J. Vreeken. "Federated Learning from Small Datasets". In: *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023.

[12] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang. "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing". In: *Proc. IEEE* 107.8 (2019), pp. 1738–1762.

[13] C. Thapa, M. A. P. Chamikara, S. Camtepe, and L. Sun. "SplitFed: When Federated Learning Meets Split Learning". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. 2022, pp. 8485–8493.

[14] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. "Adversarial machine learning". In: *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, AISec 2011*. ACM, 2011, pp. 43–58.

[15] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy, SP 2017*. IEEE Computer Society, 2017, pp. 3–18.

[16] R. Shokri and V. Shmatikov. "Privacy-Preserving Deep Learning". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1310–1321.

[17] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. "Towards Demystifying Membership Inference Attacks". In: *CoRR* abs/1807.09173 (2018). arXiv: 1807.09173.

[18] K. Chang, N. Balachandar, C. K. Lam, D. Yi, J. M. Brown, A. Beers, B. R. Rosen, D. L. Rubin, and J. Kalpathy-Cramer. "Distributed deep learning networks among institutions for medical imaging". In: *J. Am. Medical Informatics Assoc.* 25.8 (2018), pp. 945–954.

[19] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov. "Exploiting Unintended Feature Leakage in Collaborative Learning". In: *2019 IEEE Symposium on Security and Privacy, SP 2019*. 2019, pp. 691–706.

[20] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. "Learning Differentially Private Language Models Without Losing Accuracy". In: *CoRR* abs/1710.06963 (2017). arXiv: 1710.06963.

[21] E. D. Cristofaro. "An Overview of Privacy in Machine Learning". In: *CoRR* abs/2005.08679 (2020). arXiv: 2005.08679.

[22] K. Kursawe, G. Danezis, and M. Kohlweiss. "Privacy-Friendly Aggregation for the Smart-Grid". In: *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011*. Vol. 6794. Springer, 2011, pp. 175–191.

[23] E. Hoogerwerf, D. van Tetering, A. Bay, and Z. Erkin. "Efficient Joint Random Number Generation for Secure Multi-party Computation". In: *Proceedings of the 18th International Conference on Security and Cryptography, SECRYPT 2021*. SCITEPRESS, 2021, pp. 436–443.

[24] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han. "A Secure Verifiable Ranked Choice Online Voting System Based on Homomorphic Encryption". In: *IEEE Access* 6 (2018), pp. 20506–20519.

[25] A. Kiayias and M. Yung. "The Vector-Ballot e-Voting Approach". In: *Financial Cryptography, 8th International Conference, FC 2004*. Vol. 3110. Springer, 2004, pp. 72–89.

[26] C. Dwork. "Differential Privacy". In: *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 1–12.

[27] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.

[28] L. Sweeney and M. Shamos. *A multiparty computation for randomly ordering players and making random selections*. 2004.

[29] R. Martin, J. Demme, and S. Sethumadhavan. "TimeWarp: Rethinking timekeeping and performance monitoring mechanisms to mitigate side-channel attacks". In: *39th International Symposium on Computer Architecture (ISCA 2012)*. IEEE Computer Society, 2012, pp. 118–129.

[30] S. Schinzel. "An efficient mitigation method for timing side channels on the web". In: *2nd International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*. 2011, pp. 1–6.

[31] J. Ren and J. Wu. "Survey on anonymous communications in computer networks". In: *Comput. Commun.* 33.4 (2010), pp. 420–431.

[32] R. Dingledine, N. Mathewson, and P. F. Syverson. "Tor: The Second-Generation Onion Router". In: *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*. USENIX, 2004, pp. 303–320.

[33] Q. Tan, X. Wang, W. Shi, J. Tang, and Z. Tian. "An Anonymity Vulnerability in Tor". In: *IEEE/ACM Trans. Netw.* 30.6 (2022), pp. 2574–2587.

[34] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. "Hiding Routing Information". In: *Information Hiding, First International Workshop, Cambridge, UK, May 30 - June 1, 1996, Proceedings*. Vol. 1174. Lecture Notes in Computer Science. Springer, 1996, pp. 137–150.

[35] D. Chaum. "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability". In: *J. Cryptol.* 1.1 (1988), pp. 65–75.

[36] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. ACM, 2016, pp. 308–318.

[37] M. Naseri, J. Hayes, and E. D. Cristofaro. "Local and Central Differential Privacy for Robustness and Privacy in Federated Learning". In: *29th Annual Network and Distributed System Security Symposium, NDSS 2022*. 2022.

6

MACHINE LEARNING MODEL PROTECTION

Lookup arguments allow to prove that the elements of a committed vector come from a (bigger) committed table. They enable novel approaches to reduce the prover complexity of general-purpose zkSNARKs, implementing "non-arithmetic operations" such as range checks, XOR and AND more efficiently. We extend the notion of lookup arguments along two directions and improve their efficiency: (1) we extend vector lookups to matrix lookups (where we can prove that a committed matrix is a submatrix of a committed table). (2) We consider the notion of zero-knowledge lookup argument that keeps the privacy of both the sub-vector/sub-matrix and the table. (3) We present new zero-knowledge lookup arguments, dubbed cq+, zkcq+ and cq++, more efficient than the state of the art, namely the recent work by Eagen, Fiore and Gabizon named cq. Finally, we give a novel application of zero-knowledge matrix lookup argument to the domain of zero-knowledge decision tree where the model provider releases a commitment to a decision tree and can prove zero-knowledge statistics over the committed data structure. Our scheme based on lookup arguments has succinct verification, prover's time complexity asymptotically better than the state of the art, and is secure in a strong security model where the commitment to the decision tree can be malicious.

This chapter is a copy of the paper titled "Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees" by Campanelli, M., Faonio, A.*, Fiore, D., Li, T.*, and Lipmaa, H. (in alphabetical order), in *Public-Key Cryptography (PKC 2024)*, pp. 337-369, 2024.

6.1. INTRODUCTION

General-purpose zero-knowledge succinct arguments of knowledge (zkSNARKs) promise to efficiently and succinctly prove any kind of NP-statement while keeping privacy, integrity and verifiability guarantees. Thanks to their generality, a great number of real-world applications can be performed with built-in security. The two-step recipe for building a brand new zero-knowledge application typically consists of first describing the application in a low-level constraint system (for example, Rank-1 Constraint System [1] or Plonk arithemization [2]) and then use the latest fully-developed zkSNARK as *backend*. Unfortunately, most often, the *unfolded circuit* of the applications at hand becomes huge and, thus, the proving time could become unfeasible for real-world applications.

Lookup arguments [3–7] are a novel approach to reducing the size of unfolded circuits, bringing back to the real world many interesting applications. Briefly and informally, a lookup argument allows to trade *sub-circuits* evaluations for lookup into their truth tables. For example, instead of having n different sub-circuits describing the computation of a hash function in the final unfolded circuit, the protocol designer could define n different *custom gates* that perform efficient lookup operations in the truth table of such a hash function. More concretely, lookup arguments are used in current zkSNARKs for representing “non-arithmetic operations" that cannot be expressed efficiently through the finite field operations supported by the zkSNARK, such as range checks, XOR and AND (see for example [3, 8]). Very recently, the work of Arun, Setty and Thaler [9] shows how to use lookup arguments to create SNARKs for virtual-machine executions, namely a new SNARK scheme, called Jolt, that allows verification of the correct execution of a computer program specified with an assembly language. Informally, in Jolt, the truth table of each assembly instruction is encoded in a (predefined and highly structured) table. Then, lookup arguments enforce the correct instructions execution, namely checking the inputs and outputs described by their truth tables.

In this work, we advance on lookup arguments in multiple ways. We propose new lookup arguments that improve over the state of the art [4]. One of our schemes enjoys, almost for free, an extended notion of zero-knowledge, which we call fully zero-knowledge, which protects the privacy of arbitrary commitments to the tables. Orthogonally, we consider two natural extensions from vectors to matrices and give constructions for such extensions. Finally, we motivate the extensions to matrix and to fully zero-knowledge by giving a new application to privacy-preserving machine learning that crucially relies on them.

New Lookup Arguments based on cq. In a lookup argument, the prover aims to show that each coefficient of a (short) committed vector $\mathbf{f}$ of size n belongs to the (large) table $\mathbf{t}$ of size $N \gg n$. Since $N \gg n$, one of the desiderata of lookup arguments is that the prover's computation does not depend on N. Following a fast-pace line of recent works, Eagen, Fiore, and Gabizon [4] proposed an efficient lookup argument called cq (cq for *cached quotients*). Notably, cq's prover's computation is quasi-linear in n, while the proof size and verifier's computation are constant (e.g., proofs are 3840 bits, when using the standard BLS12-381 elliptic curve). In spite of appearing nearly optimal in efficiency, cq comes with two shortcomings. The first one is that it is not designed to have zero knowledge in mind. The second, more technical, one is that its use in larger protocols likely

requires additional proof elements and pairing computations.[1] In this work, we propose a new lookup argument, dubbed cq$^+$, that addresses all these shortcomings of cq and even achieves better efficiency. Namely, cq$^+$ achieves (standard) zero-knowledge at no overhead: it has the same prover's computation of cq and shorter proofs (3328 bits, and 2944 bits without ZK). Additionally, we consider two variations of cq$^+$: the first, dubbed zkcq$^+$, is fully zero-knowledge, while the second, dubbed cq^{++}, has shorter proofs. Both schemes require in verification only one pairing computation more than cq$^+$.

Lookup Arguments for Matrices. A lookup argument could be used to show that a database **f** is a selection of the rows of a database **t**. However, to naively use lookup arguments for such an application, each row of the database must be efficiently encoded in one single field element (supported by the lookup argument). We consider two natural extensions to matrices. We focus on Kate *et al.* [10] polynomial commitment (also known as KZG commitment scheme) adapted to matrices. We give two lookup arguments for matrices that internally call a lookup argument for KZG commitments. (We find this modularity useful given the current fast pace of the research on lookup arguments.)The first scheme allows proving that a committed database **f** is a selection of the rows of a committed database **t**, the second one allows proving that **f** is a selection of a projection of a database **t**.

A New Approach to Zero-Knowledge for Decision Trees. We show an application of fully zero-knowledge matrix lookup arguments to zero-knowledge for decision trees (zkDT). We improve over the framework of Zhang *et al.* [11], which showed zkSNARKs for evaluations of committed decision trees and zkSNARKs for accuracy of committed decision trees. The former kind of zero-knowledge protocols can prove that a committed decision tree T, on input a vector **x**, outputs a label v, while the latter schemes enable to validate the accuracy (namely, the ratio of true positives) of a decision tree on a given dataset.

Our framework can instantiate different kinds of statistics over committed decision trees, including evaluation and accuracy. Our design decouples the computation of the committed decision tree and the performed statistics. This allows for a plug-and-play approach. For security, we extend the notion of security from [11] considering possibly maliciously generated commitments to decision trees.

Our Contributions. We can summarize our contributions as follows:

1. A zero-knowledge lookup argument that improves the state of the art for arbitrary tables [4].

2. A construction for zero-knowledge matrix lookup argument based on zero-knowledge (vector) lookup arguments for KZG-based vector commitment.

3. A new paradigm for proving decision tree classification in zero-knowledge. We can instantiate our paradigm with our matrix lookup arguments and obtain speedups of two orders of magnitude for proving time compared to previous work.

[1] This is due to the fact that cq assumes an SRS of the same size as the table **t**, and this allows avoiding a degree check. This condition, though, is often not guaranteed (e.g., in a SNARK for constraint systems larger than such a table).

4. Strengthening the security model for zero-knowledge decision trees: we formalize a setting where the commitment to a decision tree may not be trusted.

6.1.1. TECHNICAL OVERVIEW

Our Zero-Knowledge Lookup Arguments. Similarly to cq, cq$^+$ uses the technique of logarithmic derivates of Haböck [12]. However, we diverge from cq early, introducing several novel ideas that allow us to improve on cq's efficiency. One of the differences is that, while cq uses Aurora's sumcheck [13] twice, our cq$^+$ only runs it once. Nicely, this technique allows us to kill two birds with one stone, in fact, cq$^+$ does not require any additional low-degree tests. We give a more detailed technical overview in section 6.4.1.

Matrix lookup from vector lookup. To commit to a matrix, we can commit the concatenation of the rows of the matrix. Our matrix lookup arguments label all the entries of such a vectorization with the coordinates of each cell of the sub-matrix $\mathbf{F}$ into the bigger table $\mathbf{T}$. Similarly, in the precomputation phase, they label each cell in the big table $\mathbf{T}$ with its coordinate. To prove that the k-th row of $\mathbf{F}$ appears in $\mathbf{T}$, we show that the *labelled* matrix $\mathbf{F}^* = (i_j, j, \mathbf{F}_{k,j})_{j\in[d]}$ is a sub-matrix of labeled table $\mathbf{T}^* = (i, j, \mathbf{T}_{i,j})_{i,j}$ and that $i_1 = i_2 = \cdots = i_d$ (in particular $i_j = k$), where d is the number of columns of the matrices. Notice that the first claim can be proved efficiently with a (non-succinct) matrix commitment for matrices with $N \cdot d$ rows and 3 columns following techniques from [3], while the second claim can be efficiently expressed through polynomial equations following techniques from [14]. In particular, for the first part, given a challenge $\rho \leftarrow \mathbb{F}$, the prover hashes $h(\mathbf{F}^*) = \sum_{i=1}^{3} \rho^{i-1} \cdot \mathbf{F}_i^*$ to a single column (where $\mathbf{F}_i^*$ are the columns of $\mathbf{F}^*$). Since $h(\cdot)$ is a universal hash function, if $h(\mathbf{F}^*)$ is a subvector of $h(\mathbf{T}^*)$, then with overwhelming probability, $\mathbf{F}^*$ is a submatrix of $\mathbf{T}^*$, thus reducing matrix lookup argument to vector lookup. For the second part, we notice that the first column $\mathbf{F}_1^*$ of $\mathbf{F}^*$ is a *step* function, thus we first commit to the shift of $\mathbf{F}_1^*$ and then show that the difference between the shifted column and the column $\mathbf{F}_1^*$ is a function that has zeros in well-defined positions. More details in section 6.5.2. The second scheme goes even further and proves that a matrix $\mathbf{F}$ with d' columns and $d' < d$ is a submatrix of $\mathbf{T}$. As before, we set $\mathbf{F}^* = (i, j, \mathbf{F}_{i,j})_{i\in R, j\in D}$ for subset $R = \{r_1, \ldots, r_{d'}\} \subset [N]$ and $D \subset [d]$. Additionally, using the technique of shifted polynomials, $\mathbf{F}^*_{2,id'+j} = \mathbf{F}^*_{2,(i+1)d'+j} = r_j$ for any i, j. More details in appendix A.4. Both of our compilers preserve quasi-linear running time in n thanks to the linear homomorphic property of KZG commitments.

Our Approach to ZK for Decision Trees. A decision tree is an algorithm that performs a sequence of adaptive queries reading from its input and outputs a value. At each query, the algorithm moves from a node in the tree to one of its children, and the output is defined by the label of the reached leaf. Two important parameters are the total number of nodes N_{tot} and the number of features d of the inputs. Following the work of Chen *et al.* [15], we can efficiently (although redundantly) encode a decision tree as a matrix with N_{tot} rows and $2d+1$ columns. An evaluation of a decision tree under this alternative representation consists of locating the row corresponding to the correct leaf and then showing that the input vector matches all the constraints described by such a row. Thus, we can commit to a decision tree by committing to its matrix encoding, and to prove correct evaluation, we can commit to the single row corresponding to the correct leaf

and prove with a matrix lookup argument that the committed row is indeed a leaf of the committed decision tree. Once isolated such a row, we can then prove that the input vector matches all the constraints described by the row. Notice that our strategy scales well with the number of different evaluations. In fact, to prove statements which involve multiple input vectors for the decision tree, we can commit at proving time to a matrix whose rows correspond to the entries of the leaves reached by the evaluations (instead of committing to a single row). Thanks to the efficiency property of the matrix lookup argument, the prover time complexity is independent of the size of decision trees.

Beyond a Trusted Commitment to the Tree. A malicious committer could commit to a matrix that contains a row that matches a leaf with a label, let's say, 0, and another row that matches the same leaf but where it maliciously assigns the label 1. Now, such a bogus commitment to a decision tree could allow the malicious prover to show both $\mathsf{T}(\mathbf{x}) = 0$ and $\mathsf{T}(\mathbf{x}) = 1$. The problem is that the committed matrix does not *encode* a decision tree. To solve this problem, we show a set of sufficient algebraic conditions (cf. section 6.6.2) for a matrix to *encode* a decision tree. We can check efficiently these algebraic conditions through a general-purpose zkSNARKs for R1CS (see for example [13, 16–21]). However, the number of constraints is $O(dN_{\text{tot}}^2)$, and thus the prover time complexity is quadratic in the number of nodes. The algebraic constraints we propose are essentially linear equations between matrices and Hadamard-product equations, which are the kind of equation checks performed in R1CS-based zkSNARKs. In fact, if we gave up on the privacy of the decision tree[2], we could define an R1CS circuit that depends on the tree-structure of the decision tree, and we would go down to $O(dN_{\text{tot}})$ number of constraints. We show that we can restore zero-knowledge using this approach, by privately committing to such an R1CS-like circuit and prove in zero-knowledge that the circuit belongs to a well-defined family of circuits (defined by the algebraic constraints in section 6.6.2). In particular, we use the techniques from Zapico *et al.* [6] for committing to a *basic* matrix, namely a matrix whose rows are elementary vectors, and to prove its basic-matrix structure and the permutation argument from Plonk [2] to prove the rows of the matrix are all different.

6.1.2. RELATED WORK

Lookup Arguments. Lookup arguments were introduced by Bootle, Cerulli, Groth, Jakobsen and Maller [3]. The state-of-art for lookup arguments for arbitrary tables[3] is the recent work of Eagen, Fiore, Gabizon [4] named cq and based on the technique of logarithmic derivates of Haböck [12]. cq has prover complexity proportional only to the size of the smaller vector and independent of the bigger table assuming pre-processing for table. To our knowledge, all lookup arguments with similar efficiency properties are based on the Kate *et al.* (commonly known as KZG) polynomial commitment scheme [10]. Among these, we mention Caulk+ by Posen and Kattis [5] (based on Caulk [6] by Zapico

[2] Specifically, giving up only to the privacy of the *structure of* the decision tree while keeping private the values of the thresholds and labels

[3] Recently, Setty, Thaler and Wahby [22] introduced a new lookup argument for a restricted subclass of tables. Their work is extremely efficient, and in particular more efficient than cq, for such a restricted class of tables. On the other hand, cq can handle arbitrary tables. For this reason, we refer to cq as the state-of-art for arbitrary tables.

et al.) and Baloo [7] by Zapico *et al.*. The latter work introduces the notion of Commit-and-Prove Checkable Subspace Argument (extending over [20]) that we use for our (extractable) commitment scheme (cf. section 6.6.3).

Comparison with [4]. As previously mentioned, we diverge from cq, introducing several novel ideas that allow us to improve on cq's efficiency. As the end result, cq^+'s communication is about 14% (or even 23% in a variant without the ZK) better than cq's. All other efficiency parameters of cq^+ are similar to cq's. Moreover, we propose cq^{++}, a batched variant of cq^+. cq^{++} saves 23% (or 33%, in a variant without ZK) communication compared to cq. A slight drawback of cq^{++} is that the verifier has to execute one more pairing. We emphasize that cq is already almost optimally efficient, and thus improving on it is non-trivial.

Concurrent Work. Choudhuri *et al.* [23] very recently introduced the notion of *segment lookup arguments* which, besides some syntactical differences, matches the simpler of our notions of matrix lookup arguments. Additionally, in [23], they show, in our lingo, a matrix lookup argument based on cq. Their matrix lookup argument is less efficient than ours; we defer to table 6.1 for more details. Interestingly, in the same paper, the authors build a general-purpose zkSNARK based on Plonk and matrix lookup, which they call Sublonk, showing another application for matrix lookup arguments. The main feature of Sublonk is that the prover's running time grows with the size of the *active part* of the circuit, namely the part of the circuit activated by its execution on a given instance. Sublonk makes black-box use of the underlying matrix lookup argument. Thus, we can plug in our scheme to obtain a more efficient version of Sublonk.

Privacy-Preserving Machine Learning. We focus on the related work on zero-knowledge proofs for decision trees and, more in general, for machine learning algorithms. The main related work for decision trees is the paper of Zhang *et al.* [11], where they introduce the notions of zero-knowledge proofs for decision tree predictions and accuracy. Besides decision trees, zero-knowledge proofs and verifiable computation for machine learning is a vibrant area of research (see for example [24–30]).

Comparison with [11]. Briefly, the main techniques of [11] consist of an authenticated data structure for committing to decision trees and highly-tuned R1CS circuits to evaluate the authenticated data structure in zero-knowledge. More in detail, they commit to a decision tree with a *labelled Merkle Tree* whose labelled nodes are the nodes of the decision tree. This commitment scheme is binding and hiding and allows for path openings (with proof size proportional to the length of the path). On top of this authenticated data structure, they use general-purpose zkSNARKs for R1CS to prove, for example, the knowledge of a valid opening for a path and the labelling of the leaf. While the basic ideas are simple, the paper needs to solve many technical details and presents many optimizations which are necessary to obtain a practical scheme. The *backend* general-purpose zero-knowledge scheme they use is Aurora [13]. Thanks to this choice and because of the Merkle-Tree approach, their zero-knowledge scheme has a transparent setup and is presumably post-quantum secure.

Their security model stipulates that the decision tree is adversarially chosen, but the commitment to a decision tree is honestly generated. On the other hand, in our security model, we require the commitment scheme to be extractable, thus allowing for

maliciously generated commitments. Notice that, besides improving security, our definitional choices allow for more efficient design. In fact, the (proof for the) extractable commitment is generated only once, let's say in an offline phase, while the (multiple) proofs of evaluation, in the online phase, can leverage extra security properties offered by the extractable commitment and thus faster.

For comparison with our work, we consider the extractability of their scheme for decision tree evaluation. This is not immediate: the main reason is that the witness for the zkSNARK is a single path from the root to the evaluated leaf (which could be extracted) while, to obtain our notion of extractability, it would be required to extract the full decision tree. Additionally, their authenticated data structure could allow to commit (and prove statements) to 2-fan-in direct-acyclic graphs (DAGs), which are more general than trees[4]. We believe their second scheme for the accuracy of decision trees can be proved secure in our model. In fact, proposed as an efficiency optimization, their second scheme computes a consistency check over the full decision tree. Thanks to this, we could extract the full tree from the zkSNARK. We also believe that our techniques could be integrated into theirs. Our approach separating the extractable commitment from the "online-stage" of the zero-knowledge proof could be adapted to their scheme for accuracy (thus improving its efficiency). Interestingly, by using our approach, their scheme could be interpreted as an application of a lookup argument based on [3] and [13] to decision trees. The main difference is this: our scheme runs lookup arguments over the leaves associated with the evaluation vectors, while the scheme in [11] requires lookups for paths from the root to the leaves associated with the evaluation vectors.

For other points of comparison efficiency-wise (we refer the reader to section 6.6.5 for more details), we mention that their commitments require hashing only, while ours requires multiexponentations in a group. Therefore, their commitment stage is faster than ours. Our proof size is concretely smaller (few kilobytes vs hundreds of kilobytes). To compare proving time, we start from observing the asymptotic advantages of our solutions: their prover is linear in the size of the tree and in the complexity of a hash function; ours is sublinear in all these dimensions. This results in *concretely* faster proving times *despite* the fact that our prover requires group operations and theirs only field operations. This is a consequence of removing the constants deriving from the hash function size, the sublinearity in the tree and of the efficient lookup argument instantiations[5]. Our improvements also translate to a better verification time. Our estimates show improvements of almost one order of magnitude for proving time (regardless of the underlying backend proof systems for [11]; see appendix A.1) and two orders of magnitude for verification time.

6.2. PRELIMINARIES

We denote matrices with capital and bold, for example, $\mathbf{M}$, and vectors with lowercase and bold, for example, $\mathbf{v}$. We denote with $\circ$ the Hadamard product between two ma-

[4] We believe that this does not pose any problems neither for correctness nor for soundness, as indeed, one could argue this is a feature rather than a bug.

[5] As a bottleneck, the dependency [11] has on the hash function is one that is hard to remove. Applying a hash function optimized for SNARK constraints, e.g. the one we used to experimentally run [11]—SWIFFT—*nonetheless* yields high constants in practice *regardless* of the proof system used as a backend.

trices/vectors of the same size, while $\cdot$ is reserved for the matrix-vector/vector-vector multiplication. Given two vectors $\mathbf{a},\mathbf{b}$ we define $\mathbf{a} < \mathbf{b}$ if and only if $\forall i : \mathbf{a}_i < \mathbf{b}_i$ (and similarly for $\leq$). We denote by $\|$ the concatenation by columns of two matrices. We denote by $\mathbb{F}$ a finite field, by $\mathbb{F}[X]$ the ring of univariate polynomials, and by $\mathbb{F}_{<d}[X]$ (resp. $\mathbb{F}_{\leq d}[X]$) the set of polynomials in $\mathbb{F}[X]$ of degree $< d$ (resp. $\leq d$). For any subset $S \subseteq \mathbb{F}$, we denote by $\nu_S(X) \stackrel{\text{def}}{=} \prod_{s\in S}(X-s)$ the *vanishing polynomial* of S, and by $\lambda_s^S(X)$ the *s-th Lagrange basis polynomial*, which is the unique polynomial of degree at most $|S|-1$ such that for any $s' \in S$, it evaluates to 1 if $s = s'$ and to 0 otherwise. Any multiplicative subgroup of a finite field is cyclic. Thus, given a group $\mathbb{H}$, we can find an element ω that generates the subgroup $\mathbb{H}$. For convenience, given a subgroup $\mathbb{H}$ of order n we denote with ω_n a fixed generator of $\mathbb{H}$. If $\mathbb{H} \subseteq \mathbb{F}$ is a multiplicative subgroup of order n, then its vanishing polynomial has a compact representation $\nu_{\mathbb{H}}(X) = (X^n - 1)$ and $\lambda_i^{\mathbb{H}}(X) = \nu_{\mathbb{H}}(X)\omega_n^{i-1}/(n(X-\omega_n^{i-1}))$. Both $\nu_{\mathbb{H}}(X)$ and $\lambda_i^{\mathbb{H}}(X)$ can be evaluated in $O(\log n)$ field operations. For any vector $\mathbf{v} \in \mathbb{F}^n$, we denote by $v_{\mathbb{H}}(X)$ the *low-degree encoding* (LDE) in $\mathbb{H}$ of $\mathbf{v}$, i.e., the unique degree-$(|\mathbb{H}|-1)$ polynomial such that, $v_{\mathbb{H}}(\omega_n^{i-1}) = v_i$, when the subgroup $\mathbb{H}$ is clear from the context, we simply write $v(X)$. Similarly, we consider the k-degree *randomized low-degree encoding* (RLDE) in $\mathbb{H}$ of a vector $\mathbf{v} \in \mathbb{F}^n$ to be a randomized polynomial of the form $\hat{v}_{\mathbb{H}}(X) = v_{\mathbb{H}}(X) + \nu_{\mathbb{H}}(X)\rho_v(X)$ for a random polynomial ρ_v of degree k. Sometimes, we will not explicitly mention the degree of the randomizer. In this case, the reader should assume that the degree is set to be the minimum degree necessary to keep zero-knowledge of v in the presence of evaluations (on points outside of $\mathbb{H}$) of the polynomial $\hat{v}$.

A type-3 bilinear group $\mathbb{G}$ is a tuple $(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,P_1,P_2)$. $\mathbb{G}_1,\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q. P_1,P_2 are generators of $\mathbb{G}_1,\mathbb{G}_2$. $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently-computable non-degenerate bilinear map, and there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the implicit notation $[a]_i \coloneqq aP_i$, for elements in $\mathbb{G}_i, i \in \{1,2,T\}$ and set $P_T \coloneqq e(P_1,P_2)$.

6.2.1. COMMIT-AND-PROVE SNARKS

A commitment scheme is a tuple of algorithm $\mathsf{CS} = (\mathsf{KGen},\mathsf{Com})$ where the first algorithm samples a commitment key ck and the second algorithm, upon input of the commitment key, a message p and opening material ρ, outputs a commitment c. The basic notions of security for the commitment scheme are (perfect) *hiding* and (computational) *binding*. The former property states that no (unbounded) adversary can distinguish commitments of two different messages when the opening materials are sampled at random from their domain, the latter property states that no (polynomial time) adversary, upon input of the commitment key, can find two different messages and two opening materials that commit to the same commitment.

Following Groth *et al.* [31], we define a relation $\mathcal{R}$ verifying triple $(\mathsf{pp};x;w)$. We say that w is a witness to the instance x being in the relation defined by the parameters pp when $(\mathsf{pp};x;w) \in \mathcal{R}$ (equivalently, we sometimes write $\mathcal{R}(\mathsf{pp};x;w) = 1$). For example, the parameters pp could be the description of a bilinear group, or additionally contain a commitment key for a commitment scheme or a common reference string. Whenever it is clear of the context, we will write $\mathcal{R}(x;w)$ as a shortcut for $\mathcal{R}(\mathsf{pp};x;w)$.

Briefly speaking, Commit-and-Prove SNARKs (CP-SNARKs) are zkSNARKs whose re-

lations verify predicates over commitments [32]. Given a commitment scheme CS, we consider relations $\mathcal{R}$ whose instances are of the form $x = ((\mathsf{c}_j)_{j\in[\ell]}, \hat{x})$, where we can unambiguously parse the witness $w = ((p_j)_{j\in[\ell]}, (\rho_j)_{j\in[\ell]})$ for some $\ell \in \mathbb{N}$ with $\forall j : p_j$ is in the domain of a commitment scheme CS, and such that there exists a PT relation $\hat{\mathcal{R}}$ such that let $\hat{w} = (p_j)_{j\in[\ell]}$:

$$\mathcal{R}(\mathsf{pp}; x; w) = 1 \iff \hat{\mathcal{R}}(\mathsf{pp}; \hat{x}; \hat{w}) = 1 \wedge \forall j \in [\ell] : \mathsf{c}_j = \mathsf{Com}(\mathsf{ck}, p_j, \rho_j).$$

We refer to a relation $\hat{\mathcal{R}}$ as derived above as a *Commit-and-Prove* (CP) relation. Given a CP-relation $\hat{\mathcal{R}}$ and a commitment scheme CS, we can easily derive the *associated* NP-relation $\mathcal{R}$. Instances of NP-relations may contain only commitments. Therefore, using the notation above, the instances of the associated CP-relation are empty strings ϵ, namely, $\hat{\mathcal{R}}$ is a predicate over the committed witness. To avoid cluttering the notation, in these cases, we may omit the (empty) instance and simply write $\hat{\mathcal{R}}(\mathsf{pp}, \hat{w})$.

A CP-SNARK for $\hat{\mathcal{R}}$ and commitment scheme CS is a zkSNARK for the associated relation $\mathcal{R}$ as described above. More in detail, we consider a tuple of algorithms CP = (KGen, Prove, Verify) where:

- $\mathsf{KGen}(\mathsf{ck}) \to \mathsf{srs}$ is a probabilistic algorithm that takes as input a commitment key ck for CS and it outputs $\mathsf{srs} := (\mathsf{ek}, \mathsf{vk}, \mathsf{pp})$, where ek is the evaluation key, vk is the verification key, and pp are the parameters for the relation $\mathcal{R}$ (which include the commitment key ck).
- $\mathsf{Prove}(\mathsf{ek}, x, w) \to \pi$ takes an evaluation key ek, a statement x, and a witness w such that $\mathcal{R}(\mathsf{pp}, x, w)$ holds, and returns a proof π.
- $\mathsf{Verify}(\mathsf{vk}, x, \pi) \to b$ takes a verification key, a statement x, and either accepts ($b = 1$) or rejects ($b = 0$) the proof π.

In some cases, the KGen algorithm would simply (and deterministically) re-parse the commitment key ck information. In these cases, we might omit KGen and refer to the CP-SNARK as a tuple of two algorithms.

Zero-Knowledge in the SRS (and RO) model. The zero-knowledge simulator $\mathcal{S}$ of a CP-SNARK is a stateful PPT algorithm that can operate in three modes. $(\mathsf{srs}, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}(0, 1^n, d)$ takes care of generating the parameters and the simulation trapdoor (if necessary). $(\pi, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}(1, \mathsf{st}_{\mathcal{S}}, x)$ simulates the proof for a statement x. $(a, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}(2, \mathsf{st}_{\mathcal{S}}, s)$ takes care of answering random oracle queries. The state $\mathsf{st}_{\mathcal{S}}$ is shared and updated after each operation. We define zero-knowledge similarly to [33, 34]:

Definition 6.1 (Zero-Knowledge)**.** *We say that a CP-SNARK* CP *for a CP-relation* $\hat{\mathcal{R}}$ *and commitment scheme* CS *is (perfect) zero-knowledge if there exists a PPT simulator* $\mathcal{S}$ *such that for all adversaries* $\mathcal{A}$ *and for all* $d \in \mathbb{N}$*:*

$$\Pr\left[\begin{array}{l}\mathsf{ck} \leftarrow \mathsf{CS.KGen}(1^n, d)\\ \mathsf{srs} \leftarrow \mathsf{CP.KGen}(\mathsf{ck})\\ \mathcal{A}^{\mathsf{Prove}(\mathsf{srs},\cdot,\cdot),\mathsf{RO}(\cdot)}(\mathsf{srs}) = 1\end{array}\right] \approx \Pr\left[\begin{array}{l}(\mathsf{srs}, \mathsf{st}_{\mathcal{S}}) \leftarrow \mathcal{S}(0, \mathsf{pp}_{\mathbb{G}})\\ \mathcal{A}^{\mathcal{S}_1(\cdot,\cdot),\mathcal{S}_2(\cdot)}(\mathsf{srs}) = 1\end{array}\right]$$

where $\mathcal{S}_1, \mathcal{S}_2$ are stateful (wrapper) algorithms that share their state $\mathsf{st} = (\mathsf{st}_{\mathcal{S}}, \mathcal{Q}_{\mathsf{RO}})$ *where* $\mathsf{st}_{\mathcal{S}}$ *is initially set to be the empty string, and* $\mathcal{Q}_{\mathsf{RO}}$ *is initially set to be the empty set, such that:*

- $\mathcal{S}_1(x, w)$ *denotes an oracle that first checks* $(\mathsf{pp}, x, w) \in \mathcal{R}$ *where* pp *is part of* srs *and then runs the first output of* $\mathcal{S}(1, \mathsf{st}_{\mathcal{S}}, x)$.
- $\mathcal{S}_2(s)$ *denotes an oracle that first checks if the query s is already present in* $\mathcal{Q}_{\mathsf{RO}}$ *and in case answers accordingly, otherwise it returns the first output a of* $\mathcal{S}(2, \mathsf{st}_{\mathcal{S}}, s)$. *The oracle updates* $\mathcal{Q}_{\mathsf{RO}}$ *by adding the tuple* (s, a) *to the set.*

Knowledge Soundness. Our definition of knowledge soundness is in the algebraic group model [35]. An algorithm $\mathcal{A}$ is called *algebraic* if for all group elements that $\mathcal{A}$ outputs, it additionally provides the representation relative to all previously received group elements. That is, if elems is the list of group elements that $\mathcal{A}$ has received, then for any group element z in output, the adversary must also provide a vector $\mathbf{r}$ such that $\mathsf{z} = \langle \mathbf{r}, \mathsf{elems} \rangle$. We define the notion of knowledge soundness in the algebraic model.

Definition 6.2 (Knowledge Soundness in the AGM)**.** *A CP-SNARK is knowledge extractable in the Algebraic Group Model if for any PT algebraic adversary, there exists a PT extractor $\mathcal{E}$ that receives in input the algebraic representations $\mathbf{r}_1, \ldots, \mathbf{r}_l$ of $\mathcal{A}$ and such that:*

$$\Pr\left[\begin{array}{l} \mathsf{ck} \leftarrow \mathsf{CS.KGen}(1^n, d); \mathsf{srs} \leftarrow \mathsf{CP.KGen}(\mathsf{ck}); \\ (x, \pi, \mathbf{r}_1, \ldots, \mathbf{r}_l) \leftarrow \mathcal{A}(\mathsf{srs}); w \leftarrow \mathcal{E}(\mathsf{srs}, \mathbf{r}_1, \ldots, \mathbf{r}_l) \\ \mathsf{Verify}(\mathsf{srs}, x, \pi) \wedge \neg \mathcal{R}(\mathsf{pp}, x, w) \end{array}\right] \leq \mathsf{negl}(n)$$

Indexed Relations and Universal CP-SNARKs. We extend the notion of relations to indexed relations [36]. We define a PT indexed relation $\mathcal{R}$ verifying tuple (pp, ind, x, w). We say that w is a witness to the instance x being in the relation defined by the pp and index ind when $(\mathsf{pp}, ind, x, w) \in \mathcal{R}$ (equivalently, we sometimes write $\mathcal{R}(\mathsf{pp}, ind, x, w) = 1$).

Briefly, we say that a CP-SNARK is *universal* if there exists a deterministic algorithm Derive that takes as input an srs and an index ind, and outputs a specialized reference string $\mathsf{srs}_{ind} = (\mathsf{vk}_{ind}, \mathsf{ek}_{ind})$ where vk_{ind} is a succinct verification key and ek_{ind} is a proving key for such an index. Moreover, we require that the verifier Verify (resp. the P) of a Universal CP-SNARK takes as additional input the specialized verification key vk_{ind} (resp. the specialized ek_{ind}). We refer to appendix A.2 for more details.

6.2.2. EXTRACTABLE COMMITMENT SCHEMES

An extractable commitment scheme for a domain $\mathcal{D} = \{\mathcal{D}_n\}_n$ is a commitment scheme equipped with a CP-SNARK that proves the knowledge of an opening of the commitments.

Definition 6.3. *Given a domain $\mathcal{D}$,* $\mathsf{CS} = (\mathsf{KGen}, \mathsf{Com}, \mathsf{VerCom})$ *is an extractable commitment scheme for the domain $\mathcal{D}$ if there exist two algorithms* $\mathsf{Com}', \mathsf{Prove}'$ *such that* $\mathsf{Com}(\mathsf{ck}, p, \rho)$ *executes (1)* $\mathsf{c} \leftarrow \mathsf{Com}'(\mathsf{ck}, p, \rho)$ *and (2)* $\pi \leftarrow \mathsf{Prove}'(\mathsf{ck}, \mathsf{c}, (p, \rho))$ *and outputs* (c, π), *and* $(\mathsf{Prove}', \mathsf{VerCom})$ *is a CP-SNARK for the commitment scheme* $(\mathsf{KGen}, \mathsf{Com}')$ *and for the CP-Relation $\hat{\mathcal{R}}_{\mathsf{open}}$ defined below:*

$$\hat{\mathcal{R}}_{\mathsf{open}} = \{\mathsf{pp}; \varepsilon; p : p \in \mathcal{D}_n\}.$$

6.2.3. POLYNOMIAL, VECTOR AND MATRIX COMMITMENT SCHEMES

We use the KZG polynomial commitment scheme of [10] described below:

$\mathsf{KGen}(1^n, d_1, d_2)$ samples a type-3 pairing group with security level n and outputs commitment key $\mathsf{ck} := ((\left[s^i\right]_1)_{i\in[d_1]}, (\left[s^i\right]_2)_{i\in[d_2]},)$ for random secrets $s \in \mathbb{Z}_q$.

$\mathsf{Com}(\mathsf{ck}, p)$ outputs $\left[p(s)\right]_1$.

We notice that the above commitment scheme is not hiding and it is extractable[6] for the domain of polynomial of degree d_1 in the algebraic group model of [35] under the power discrete logarithm assumption (PDL), which informally states that find s is hard given a freshly sampled commitment key, see definition A.2 for details. The commitment scheme allows for a very efficient CP-SNARK $\Pi_{\mathsf{eval}} = (\mathsf{Prove}_{\mathsf{eval}}, \mathsf{Verify}_{\mathsf{eval}})$ for the CP-relation $\hat{\mathcal{R}}_{\mathsf{eval}} = \{(x, y; p)\colon p(x) = y\}$. In particular, the prover $\mathsf{Prove}_{\mathsf{eval}}$ upon input the SRS ck, an instance $(\left[p(s)\right]_1, x, y)$ and the witness p, computes the unique polynomial w such that the equation below holds and outputs $[w(s)]_1$ as its proof:

$$p(X) = w(X) \cdot (X - x) + y.$$

On the other hand, the verifier $\mathsf{Verify}_{\mathsf{eval}}$ upon input the SRS ck, an instance (c, x, y) and a proof π, checks $e(\mathsf{c} - \left[y\right]_1, [1]_2) = e(\pi, [s - x]_2)$.

Vector and Matrix commitment schemes. From a polynomial commitment scheme, we can define a *vector* commitment. Specifically, let $\mathbb{H}$ be multiplicative subgroup of $\mathbb{F}$ with order N, and let ω_N be a fixed generator of $\mathbb{H}$. We can commit to vector $\mathbf{v}$ by committing to the low-degree encoding of v over $\mathbb{H}$. Namely, $[v_{\mathbb{H}}(s)]_1$ is a commitment to $\mathbf{v}$. The commitment key should additionally contain the description of the subgroup $\mathbb{H}$ to allow for verification. Notice that such a commitment scheme is not hiding. We can make it hiding by committing to a RLDE of v over $\mathbb{H}$ instead of its LDE. We can easily adapt the CP-SNARK for $\mathcal{R}_{\mathsf{eval}}$ to spot-opening of a committed vector.

We define the *vectorization* of a matrix $\mathbf{M} \in \mathbb{F}^{n\times d}$ to be the vector $\bar{\mathbf{m}} \in \mathbb{F}^{n\cdot d}$ which is the concatenation of the rows of $\mathbf{M}$. Namely, for any $i \in [n], j \in [d]$, we define $\bar{\mathbf{m}}_{d\cdot i+j} = \mathbf{M}_{i,j}$. To commit to a matrix $\mathbf{M}$, we commit to its vectorization $\bar{\mathbf{m}}$. Notice that, additionally, the commitment key should contain the values n and d, and the subgroup $\mathbb{H}$ should be of cardinality $n \cdot d$.

6.3. ZERO-KNOWLEDGE MATRIX LOOKUP ARGUMENTS

Given two vectors $\mathbf{f}, \mathbf{t}$, we say that $\mathbf{f}$ is a *sub-vector* of $\mathbf{t}$ if there exits a (multi) set $K = \{k_1, \ldots, k_n\}$ such that $\mathbf{f}_j = \mathbf{t}_{k_j}$ for any j. We write $\mathbf{f} \prec \mathbf{t}$ to denote that $\mathbf{f}$ is a sub-vector of $\mathbf{t}$. Notice we diverge from the usual notion of sub-vector. Namely, we assume that a sub-vector $\mathbf{f}$ may contain multiple copies of an element in $\mathbf{t}$ and, moreover, any permutation

[6] As argued in [16], we can define a *vacuous* CP-SNARK for opening in the AGM where the prover does nothing and the verifier checks that the commitment is a valid group element. However, Lipmaa et al. [37] recently defined AGMOS, a more realistic variant of the AGM where the algebraic adversary can obliviously sample group elements. They pointed out that KZG is only extractable after the prover has successfully opened the commitment at some point. In this case, such a vacuous CP-SNARK is not sufficient. We leave it to further work to prove the security of our protocols in AGMOS.

of $\mathbf{f}$ is a sub-vector of $\mathbf{t}$. We extend the notion of sub-vectors to matrices. We say that a matrix $\mathbf{F} \in \mathbb{F}^{n\times d}$ is a (rows) sub-matrix of a matrix $\mathbf{T} \in \mathbb{F}^{N\times d}$ if $\mathbf{F}$ parsed as a $\mathbb{F}^d$-vector of length n is a sub-vector of $\mathbf{T}$ parsed as a $\mathbb{F}^d$-vector of length N. In other words, $\mathbf{F}$ is a matrix whose rows are also rows in $\mathbf{T}$. Similarly, given a multi set $K = \{k_1, \ldots, k_l\}$ we can define the sub-matrix $\mathbf{F}_{|K}$ as the sub-matrix of $\mathbf{F}$ which j-th row is the row $\mathbf{F}_{k_j}$. Notice that our notion of sub-matrix is not standard. Besides the differences mentioned for the notion of sub-vector, we consider the special case where the number of columns of $\mathbf{F}$ and $\mathbf{T}$ are the same. This is sufficient for our application. However, for completeness, in appendix A.4.1, we consider the more general case where $\mathbf{F}$ may be a selection of a projection of $\mathbf{T}$. We call the latter the rows-columns sub-matrix relationship. We consider the following indexed CP-relation, where we will refer to $\mathbf{T}$ as the table and to $\mathbf{F}$ as the sub-vector (or sub-matrix):

$$\hat{\mathcal{R}}_{\mathsf{zklookup}} := \{\mathsf{pp}; (N, d, n); \varepsilon; (\mathbf{T}, \mathbf{F}) : \mathbf{F} \prec \mathbf{T}, |\mathbf{T}| = N \times d, |\mathbf{F}| = n \times d\}, \tag{6.1}$$

Previous work focuses on $d = 1$, namely the lookup argument for vector commitments, where the table $\mathbf{T}$ is public. Moreover, some of the previous work did not focus on zero-knowledge. Namely, previous work focused on (ZK or not) CP-SNARKs for the following CP-relation:

$$\hat{\mathcal{R}}_{\mathsf{lookup}} := \{\mathsf{pp}; (\mathbf{t}, n); \varepsilon; \mathbf{f} : \mathbf{f} \prec \mathbf{t}, |\mathbf{f}| = n\}. \tag{6.2}$$

A *fully* zero-knowledge lookup argument for a commitment scheme CS is a CP-SNARK for the CP-relation $\hat{\mathcal{R}}_{\mathsf{zklookup}}$ and for the commitment scheme CS. We use the adjective fully zero-knowledge to distinguish our definition from the definition from previous work. State-of-the-art lookup arguments have prover time complexity independent of the length of the table and quasi-linear (or even linear) on the length of the sub-vector. To obtain such a property, all the lookup arguments for arbitrary tables in previous work precompute the table $\mathbf{T}$, producing auxiliary material that is then used during the proving phase. Thus, using the notational framework of Universal SNARK, the precomputation is handled by the Derive algorithm (since $\mathbf{t}$ is in the index).

Definition 6.4. *A tuple of algorithm* CP = (KGen, Derive, Prove, Verify) *is a lookup argument for a commitment scheme* CS *if (1)* CP *forms a CP-SNARK for* $\hat{\mathcal{R}}_{\mathsf{lookup}}$ *and* CS*, (2)* Derive *is a* $\mathbb{F}$*-linear function (with respect to the proving key in its output) and the commitment scheme is linearly homomorphic and (3)* Prove *has running time* $\mathsf{poly}(n, n)$.

We define an additional algorithm Preproc to handle our stronger privacy requirement. Similarly to Derive, the algorithm Preproc performs an offline preprocessing — both algorithms are necessary *only* for speeding up the proving and verification algorithms. The difference is that Derive works over *public* information, meanwhile Preproc works over *private* information[7].

[7] Alternatively, one could define one single algorithm Derive that handles both public and private data. In this case, one needs to redefine the Universal SNARK's framework to handle zero knowledge correctly. Our definition instead is only functional as we require that Preproc, Prove form a two-step prover algorithm for a Universal SNARK.

Definition 6.5. *A tuple of algorithm* $\mathsf{CP} = (\mathsf{KGen}, \mathsf{Derive}, \mathsf{Preproc}, \mathsf{Prove}, \mathsf{Verify})$ *is a fully zero-knowledge lookup argument for a matrix commitment scheme* CS *if (1)* $(\mathsf{KGen}, \mathsf{Derive}, \mathsf{Prove}', \mathsf{Verify})$ *forms a CP-SNARK for* $\hat{\mathcal{R}}_{\mathsf{zklookup}}$ *and* CS *where* Prove' *is the algorithm that upon witness* $(\mathbf{T}, \mathbf{F}, \rho_T, \rho_M)$ *such that* $\mathbf{T}_{|K} = \mathbf{F}$ *first runs* $(\mathsf{aux}_j)_{j\in[N]} \leftarrow \mathsf{Preproc}(\mathsf{srs}, \mathbf{T}, \rho_T)$ *and then runs* Prove *with witness* $(\mathbf{F}, \rho_M, (\mathsf{aux}_j)_{j\in K})$*; (2)* $\mathsf{Preproc}$ *is a* $\mathbb{F}$*-linear function and the commitment scheme is linearly homomorphic and (3)* Prove *has running time* $\mathsf{poly}(nd, n)$.

6.4. OUR NEW ZERO-KNOWLEDGE LOOKUP ARGUMENTS

In this section, we present our new lookup arguments for KZG-based vector commitments. Let the commitment $\mathsf{c}_{\mathbf{t}}$ and $\mathsf{c}_{\mathbf{f}}$, to the vectors $\mathbf{t}$ and $\mathbf{f}$ respectively, be KZG commitments to randomized low-degree encodings of $\mathbf{t}$ and $\mathbf{f}$. We denote these polynomials $\mathsf{T}(X)$ and $\mathsf{F}(X)$, respectively. Since $\mathbf{t}$ and $\mathbf{f}$ have different sizes, we interpolate them over two multiplicative subgroups of $\mathbb{F}$: $\mathbb{K}$ of order N and $\mathbb{H}$ of order $n \leq N$. In our construction, we need $n \mid N$; however, this usually holds in practice where both n and N are powers of two. Hence, we have

$$\mathsf{T}(X) := \sum_{j=1}^{N} \mathbf{t}_j \lambda_j^{\mathbb{K}}(X) + \rho_{\mathsf{T}} \cdot \nu_{\mathbb{K}}(X), \quad \mathsf{F}(X) := \sum_{i=1}^{n} \mathbf{f}_i \lambda_i^{\mathbb{H}}(X) + \rho_{\mathsf{F}}(X) \cdot \nu_{\mathbb{H}}(X)$$

Above, $\rho_{\mathsf{F}}(X)$ is a random polynomial of degree $< \mathsf{b}_F$ so that $\mathsf{c}_{\mathbf{f}} = [\mathsf{F}(s)]_1$ is perfectly hiding. Furthermore, our lookup arguments work (and are zero-knowledge) for any choice of $\mathsf{b}_F \geq 0$; this property matters whenever the commitment $\mathsf{c}_{\mathbf{f}}$ is generated by other protocols with their own zero-knowledge requirements (e.g., $\mathsf{c}_{\mathbf{f}}$ may come from a SNARK construction where b_F is carefully set to meet the number of leaked evaluations of $\mathsf{F}(X)$ in that protocol). Our lookup arguments achieve zero knowledge without leaking additional evaluations of $\mathsf{F}(X)$.

On the other hand, if $\rho_{\mathsf{T}} \leftarrow\!\!\$\ \mathbb{F}$ is a random field element, then $\mathsf{c}_{\mathbf{t}} = [\mathsf{T}(s)]_1$ is a perfectly hiding commitment to $\mathbf{t}$. Otherwise, if $\rho_{\mathsf{T}} = 0$, we capture the case of public tables (that is the common use case of lookup arguments).

We use the following lemma from [12].

Lemma 6.1 (Set inclusion, [12])**.** *Let* $\mathbb{F}$ *be a field of characteristic* $p > N$*, and suppose that* $(a_i)_{i=1}^N, (b_i)_{i=1}^N$ *are arbitrary sequences of field elements. Then* $\{a_i\} \subseteq \{b_i\}$ *as sets (with multiples of values removed), if and only if there exists a sequence* $(m_i)_{i=1}$ *of field elements from* $\mathbb{F}_p \subseteq \mathbb{F}$ *such that*

$$\sum_{i=1}^{N} \frac{1}{X - a_i} = \sum_{i=1}^{N} \frac{m_i}{X - b_i} \tag{6.3}$$

in the function field $\mathbb{F}(X)$*. Moreover, we have equality of the sets* $\{a_i\} = \{b_i\}$*, if and only if* $m_i \neq 0$*, for every* $i = 1, \ldots, N$.

Roadmap. For the sake of presentation, we first describe our main lookup argument cq^+, which works for a public table $\mathbf{t}$, thus meeting definition 6.4. This protocol is fully described in Fig. 6.1 and explained in the next section. Next, we discuss an optimized variant, cq^{++}. Finally, in Section 6.4.2 we show how to obtain the protocol meeting the fully zero-knowledge notion of definition 6.5.

6.4.1. cq$^+$ LOOKUP ARGUMENT

For ease of exposition, we present our protocol as a public coin interactive argument. We can compile it into a CP-SNARK using the Fiat-Shamir heuristic.

Setup. We assume a universal srs $= ((\left[s^j\right]_1)_{j=0}^{N_1}, (\left[s^j\right]_2)_{j=0}^{N_2})$ for any $N_1 \geq N + \max(\mathsf{b}_F, 1) - 1$ and $N_2 \geq N + \max(\mathsf{b}_F, 1) + 1$, where b_F is the degree of the randomization polynomial $\rho_{\mathsf{F}}(X)$ explained earlier.

Round 1. Our interactive lookup protocol cq$^+$ starts the same as cq [4]. Namely, based on theorem 6.1, the prover computes the multiplicities vector $\mathbf{m}$ such that $\sum_{j=1}^{N} \frac{m_j}{\mathbf{t}_j + X} = \sum_{i=1}^{n} \frac{1}{\mathbf{f}_i + X}$, and sends to the verifier a commitment $[m(s)]_1$ to a randomized low-degree encoding $m(X)$ of $\mathbf{m}$ over $\mathbb{K}$.

Round 2. The verifier sends a random challenge β. At this point, the goal of the prover is to convince the verifier that

$$\sum_{j=1}^{N} \frac{m_j}{\mathbf{t}_j + \beta} = \sum_{i=1}^{n} \frac{1}{\mathbf{f}_i + \beta} \tag{6.4}$$

which, by Schwartz-Zippel, implies the polynomial identity over $\mathbb{F}[X]$ and thus $\mathbf{f} \prec \mathbf{t}$ by Lemma 6.1. To this end, the prover commits to randomized low-degree encodings of the two vectors containing the terms of the two sums, i.e.,

$$A(X), B(X) \text{ s.t. } A_j = A(\omega_N^{j-1}) = \frac{m_j}{\mathbf{t}_j + \beta} \quad \text{and} \quad B_i = B(\omega_n^{i-1}) = \frac{1}{\mathbf{f}_i + \beta} \ . \tag{6.5}$$

In order to prove the well-formedness of $A(X)$ and $B(X)$, as in cq, the prover commits to the polynomials $Q_A(X) = (A(X)(\mathsf{T}(X) + \beta) - m(X))/\nu_{\mathbb{K}}(X)$ and $Q_B(X) = (B(X)(\mathsf{F}(X) + \beta) - 1)/\nu_{\mathbb{H}}(X)$. As we discuss later, we compute a commitment to $Q_A(X)$ using the cached quotients technique of [4] to meet the efficiency requirement (3) of definition 6.4.

From this point, our protocol diverges from cq. At this point of the protocol, cq would proceed by applying Aurora's univariate sumcheck on both $A(X)$ and $B(X)$ to prove the correctness of results $A(0) = \sum_j A(\omega_N^{j-1})/N$ and $B(0) = \sum_i B(\omega_n^{i-1})/n$ and then the verifier would check that the results are equal.

In cq$^+$, we instead apply Aurora's univariate sumcheck on a scaled sum of $A(X)$ and $B(X)$ and prove that the result is zero. More precisely, we define $C(X) := A(X) - \vartheta^{-1} B(X)\mathsf{z}(X)$ where we denote $\vartheta := N/n$ and $\mathsf{z}(X) := \nu_{\mathbb{K}\setminus\mathbb{H}}(X)$ and use the following lemma (see Appendix A.3 for its proof).

Lemma 6.2. $\sum_{j=0}^{N} A(\omega_N^{j-1}) = \sum_{i=0}^{n} B(\omega_n^{i-1})$ *iff* $\sum_{j=0}^{N} C(\omega_N^{j-1}) = 0$.

The lemma relies on the observation that the polynomial $\Delta(X) := \vartheta^{-1} B(X)\mathsf{z}(X)$ encodes over $\mathbb{K}$ the same vector encoded by $B(X)$ over $\mathbb{H}$, i.e., $\left(\frac{1}{\mathbf{f}_i + \beta}\right)_i$, but in different positions; while in the rest of positions it encodes zeros. Thus, $\sum_{j=0}^{N} \Delta(\omega_N^{j-1}) = \sum_{i=0}^{n} B(\omega_n^{i-1})$. Moreover, multiplying $B(X)$ by $\mathsf{z}(X)$ gives us for free a low-degree test on $B(X)$.

Thus, towards proving (6.4), we use Aurora's sumcheck on $C(X)$ to show

$$\exists R_C(X) \in \mathbb{F}_{\leq N-2}[X],\ Q_C(X) \text{ s.t. } C(X) = R_C(X)X + Q_C(X)\nu_{\mathbb{K}}(X) \ . \tag{6.6}$$

However, we do not send commitments to these two polynomials but use alternative techniques that allow us to obtain both zero knowledge and an efficient degree check on $R_C(X)$. More precisely, to obtain zero-knowledge, we use the sparse ZK sumcheck technique from Lunar [16]: the prover commits to a polynomial $S(X) := R_S X + \rho_S v_{\mathbb{K}}(X)$, with the idea that in the next round we perform a sumcheck on $C(X) + \eta^2 S(X)$, for a random challenge η to be chosen by the verifier in the following round. Actually, although for ease of expositions we introduced the use of $S(X)$ here; this polynomial is computed and committed as $[S(s)]_1$ in round 1. In summary, in round 2, the prover sends $[A(s), B(s), Q_B(s)]_1$.

Round 3. The verifier sends random challenges γ, η. In this round, the prover's goal is to show that

$$A(X)(\mathsf{T}(X) + \beta) - m(X) = Q_A(X) v_{\mathbb{K}}(X), \tag{6.7}$$

$$B(X)(\mathsf{F}(X) + \beta) - 1 = Q_B(X) v_{\mathbb{H}}(X), \tag{6.8}$$

$$A(X) - \vartheta^{-1} B(X)\mathsf{z}(X) + \eta^2 S(X) - (R_C(X) + \eta^2 R_S)X = Q_{C,S}(X) v_{\mathbb{K}}(X) \tag{6.9}$$

where $Q_{C,S}(X) = Q_C(X) + \eta^2 \rho_S$ in (6.9). To prove equation (6.7), we use the cached quotient technique of [4] to compute a commitment $[Q_A(s)]_1$ using n scalar group multiplications (see below).

To prove equation (6.8), notice that we already sent $Q_B(X)$; thus, using a linearization trick and random point evaluations, we set $B_\gamma = B(\gamma)$ and we show $B(X)$ evaluates to B_γ on γ, $D(X) := B_\gamma(\mathsf{F}(X) + \beta) - 1 - Q_B(X) v_{\mathbb{H}}(\gamma)$ evaluates at 0 on γ. We batch these claims using the verifier's challenge η. Namely, we send the KZG-evaluation proof $P(X) := ((B(X) - B_\gamma) + \eta D(X))/(X - \gamma)$.

To prove equation (6.9), we apply a novel idea that allows obtaining, for free, a degree check on $R_C(X)$. We set the polynomial $U(X) = (X^\mu - 1)$ where $\mu = N_1 - N + 2$ and ask the prover to send $R^*_C(X) = (R_C(X) + \eta^2 R_S)U(X)$. To balance this, we multiply the rest of equation (6.9) by $U(X)$, obtaining

$$(A(X) - \vartheta^{-1} B(X)\mathsf{z}(X) + \eta^2 S(X))U(X) - R^*_C(X)X = Q_{C,S}(X) v_{\mathbb{K}}(X) U(X) \tag{6.10}$$

To further optimize this, we batch equations (6.7) and (6.10) by using the verifier's random challenge η (and multiplying (6.7) by $U(X)$), finally obtaining:

$$\begin{aligned} &A(X) \cdot \mathsf{T}(X)U(X) + ((\beta + \eta)A(X) - m(X) + \eta^2 S(X)) \cdot U(X) \\ &\qquad - \frac{\eta}{\vartheta} B(X) \cdot \mathsf{z}(X)U(X) - Q(X) v_{\mathbb{K}}(X) U(X) = \eta R^*_C(X) \cdot X \end{aligned} \tag{6.11}$$

The idea of this batching is that after multiplying (6.7) by $U(X)$, both equations aim to prove that the left-hand side is divisible by $v_{\mathbb{K}}(X)$ and thus we can send a single quotient polynomial $Q(X) = Q_A(X) + \eta Q_C(X) + \eta^2 \rho_S$.

To summarize, in round 3, the prover sends $\left[P(s), R^*_C(s), Q(s)\right]_1$ and B_γ.

Verification. The verifier proceeds as described in Verify of Fig. 6.1. The verification item (ii) is a standard technique to check the batched evaluation proof $[P(s)]_1$. The verification item (i) instead implements the check of eq. (6.11) using pairings. Doing this

requires the verifier to have in the verification key the $\mathbb{G}_2$ elements $[\mathsf{T}(s)U(s)]_2$ as well as $[U(s), \mathsf{z}(s)U(s), \nu_{\mathbb{K}}(s)U(s)]_2$. Therefore, we let Derive compute all these elements and include them in the verification key.

Prover efficiency. We discuss how the prover algorithm can be implemented with $O(n)$ scalar multiplications in $\mathbb{G}_1$ and $O(n\log n)$ $\mathbb{F}$ operations. First, one can easily see that by preprocessing the computation of the elements $\left[\lambda_j^{\mathbb{K}}(s)\right]_1$ and $[\nu_{\mathbb{K}}(s)]_1$ and by using the n-sparsity of $\mathbf{m}$, it is possible to compute $[m(s), A(s)]_1$ using $2(n+1)$ scalar multiplications. Computing $Q_B(X)$ is the only step that requires time $O(n\log n)$ (in field operations). Computing $[B(s), Q_B(s), P(s)]_1$ requires $\approx 3n$ scalar multiplications.

Computing the commitments $\left[R_C^*(s)\right]_1$ and $[Q_A(s)]_1$ with $\approx 2n$ and n scalar multiplications, respectively, can be achieved thanks to the cached quotients and, again, the sparseness of $\mathbf{m}$. Following [4], in Derive for $\mathbf{t}$, we compute and store

$$\left[Q_j(s)\right]_1 \text{ where } Q_j(X) := \frac{(\mathsf{T}(X)-\mathbf{t}_j)\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} \ .$$

Then, we use this auxiliary input to compute, with $n+1$ scalar multiplications,

$$[Q_A(s)]_1 \leftarrow \textstyle\sum_{m_j\neq 0} A_j \left[Q_j(s)\right]_1 + \left[\rho_A(T(s)+\beta)-\rho_m\right]_1 \ . \tag{6.12}$$

The correctness of $Q_A(s)$ is due to

$$\begin{aligned}
\textstyle\sum_{j=1}^N A_j Q_j(X) &= \textstyle\sum_{j=1}^N \frac{A_j(\mathsf{T}(X)-\mathbf{t}_j)\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} \\
&= \textstyle\sum_{j=1}^N \frac{A_j(\mathsf{T}(X)+\beta)\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} - \sum_{j=1}^N \frac{A_j(\mathbf{t}_j+\beta)\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} \\
&\overset{(6.5)}{=} (\mathsf{T}(X)+\beta)\textstyle\sum_{j=1}^N \frac{A_j\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} - \sum_{j=1}^N \frac{m_j\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} \\
&= \tfrac{(A(X)-\rho_A\nu_{\mathbb{K}}(X))(\mathsf{T}(X)+\beta)-m(X)+\rho_m\nu_{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)} \\
&= Q_A(X) - \rho_A(T(X)+\beta)+\rho_m \ .
\end{aligned}$$

Using a similar technique, in Derive we can precompute $\left[(r_j^{\mathbb{K}}(s))_{j=1}^N, (r_i^{\mathbb{H}}(s))_{i=1}^n\right]_1$ where $\left\{r_j^{\mathbb{K}}(X) = \frac{\lambda_j^{\mathbb{K}}(X)-\lambda_j^{\mathbb{K}}(0)}{X}U(X)\right\}_{j\in[N]}$, and $\left\{r_i^{\mathbb{H}}(X) = \frac{\lambda_i^{\mathbb{H}}(X)\mathsf{z}(X)-\lambda_i^{\mathbb{H}}(0)}{X}U(X)\right\}_{i\in[n]}$, and use them to compute $\left[R_C^*(s)\right]_1$ in $2n$ scalar multiplications.

Thus, the prover's computation is dominated by $8n$ scalar multiplications, which was also the case in cq that did not achieve zero-knowledge and assumed $A(X)$ to be of degree $< N$.

cq^{++}: a variant with a shorter proof. We can further optimize cq$^+$ by applying one more batching technique that consists of sending a single group element $[P^*(s)]_1 = \left[P(s)+R_C^*(s)\right]_1$ and in merging the two verification equations ((i)) and ((ii)) as follows:

$$\begin{aligned}
&e\left([A(s)]_1, \left[\mathsf{T}(s)U(s)(s-\gamma)\right]_2\right)\cdot e\left(\left[(\beta+\eta)A(s)-m(s)+\eta^2 S(s)\right]_1, \left[U(s)(s-\gamma)\right]_2\right)\cdot \\
&\quad e\left(\tfrac{\eta}{\vartheta}[B(s)]_1, \left[\mathsf{z}(s)U(s)(s-\gamma)\right]_2\right)^{-1}\cdot e\left([Q(s)]_1, \left[\nu_{\mathbb{K}}(s)U(s)(s-\gamma)\right]_2\right)^{-1}\cdot \\
&\quad e\left(\eta\left[B(s)+\eta D(s)-B_\gamma\right]_1, [s]_2\right) = e\left(\eta\left[P^*(s)\right]_1, \left[s(s-\gamma)\right]_2\right) \ .
\end{aligned}$$

This change also requires some small changes. First, we require in the srs to have $N_2 \geq N + \max(\mathsf{b}_F, 1) + 2$. Second, the verification key $\mathsf{vk}_{N,n}$ computed by Derive must include $\left[(s^k U(s), s^k \mathsf{z}(s) U(s), s^k \nu_{\mathbb{K}}(s) U(s))_{k=0}^{1}\right]_2$. Third, the table-dependent verification key for $\mathbf{t}$ should include $\left[(s^k \mathsf{T}(s) U(s))_{k=0}^{1}\right]_2$.

Overall efficiency. Assume that we use a standard curve like BLS12-381, where elements of $\mathbb{G}_1$ (resp., $\mathbb{F}$) are $\mathfrak{g}_1 = 384$ (resp., $\mathfrak{f} = 256$) bits long. Then, in cq^+, the communication is $8\mathfrak{g}_1 + 1\mathfrak{f}$ (3328 bits) and in cq^{++}, $7\mathfrak{g}_1 + 1\mathfrak{f}$ (2944 bits). The prover executes $\approx 8n$ scalar multiplications. Verifier has to execute 5 pairings in cq^+ or 6 in cq^{++}. Importantly, two or three of the pairings are with the standard $\mathbb{G}_2$ element (depending on the variant, $[1, x]_2$ or $\left[1, x, x^2\right]_2$). Hence they can be batched with other pairings in the master protocol and essentially come for free.

If one does not wish ZK, we can remove $[S(s)]_1$ from the argument, and proof size is $7\mathfrak{g}_1 + 1\mathfrak{f}$ (2944 bits) in cq^+, and $6\mathfrak{g}_1 + 1\mathfrak{f}$ (2560 bits) in cq^{++}.

To compare, in cq [4] (that is not ZK), the communication is $8\mathfrak{g}_1 + 3\mathfrak{f}$ (3840 bits), the prover's computation is $\approx 8n$ scalar multiplications, and the verifier has to execute 5 pairings. Hence, even cq^+ (*with* ZK) has better communication than cq (*without* ZK) while having the same cost in the rest of the parameters.

Security. In the following theorem, we argue the security of cq^+ (see Appendix A.3 for the proof and the definition of the Power Discrete Logarithm (PDL) assumption); the proof of cq^{++} is very similar.

Theorem 6.3. *The protocol* cq^+ *from fig. 6.1 is a lookup argument according to definition 6.4. Specifically,* cq^+ *is knowledge-sound in the AGM and ROM under the* (N_1, N_2)*-PDL assumption (see definition A.2), and, furthermore, the protocol is zero-knowledge.*

6.4.2. OUR FULLY ZERO-KNOWLEDGE LOOKUP ARGUMENT

In this setting we have $\mathsf{T}(X) = \sum_{j=1}^{N} = \mathbf{t}_j \lambda_j^{\mathbb{K}}(X) + \rho_{\mathsf{T}} \cdot \nu_{\mathbb{K}}(X)$ where $\rho_{\mathsf{T}} \leftarrow\!\$\ \mathbb{F}$ and $\mathsf{c}_{\mathbf{t}} = [\mathsf{T}(s)]_1$. We need only slight modifications to turn cq^+ to a fully zero-knowledge lookup argument. We refer to the modified lookup argument as zkcq^+, formally described in fig. A.2 in appendix A.3.2. First, we defer, from Derive to Preproc, the computation of all the *table-dependent* group elements. Namely, $\mathsf{Preproc}(\mathsf{srs}, \mathbf{t}, \rho_{\mathsf{T}})$ computes $\left(\left[Q_j(s)\right]_1\right)_{j=1}^{N}$ and $\tilde{\mathsf{c}}_{\mathbf{t}} \leftarrow [\mathsf{T}(s)U(s)]_2$. The latter group element is included as part of the proof at proving time by the algorithm Prove. As consequence, Verify needs to additionally run the pairing check $e([1]_1, \tilde{\mathsf{c}}_{\mathbf{t}}) = e(\mathsf{c}_{\mathbf{t}}, [U(s)]_2)$ to verify the well-formedness of the commitment $\mathsf{c}_{\mathbf{t}}$. In the proof of knowledge soundness, this check allows us to ensure that the polynomials extracted from $\mathsf{c}_{\mathbf{t}}$ and $\tilde{\mathsf{c}}_{\mathbf{t}}$ are of the form $\mathsf{T}^*(X)$ and $\mathsf{T}^*(X)U(X)$ for some $\mathsf{T}^*(X)$; thus, after verifying this we can apply virtually the same proof of Theorem 6.3.

6.5. OUR MATRIX LOOKUP ARGUMENT

We show a compiler from a fully zero-knowledge vector lookup argument for KZG-based vector commitment to a fully zero-knowledge matrix lookup for the (succinct) KZG-based matrix commitment from section 6.2.3. The same construction applies for lookup argument as in definition 6.4.

Derive$(\mathsf{srs}, \mathbf{t}, n)$**:** // Assume that $|\mathbf{t}| = N = |\mathbb{K}|$, $n = |\mathbb{H}|$ and $n \mid N$, $\mathsf{srs} = (\left[(s^j)_{j\in[N_1]}\right]_1, \left[(s^j)_{j\in[N_2]}\right]_2)$ for any $N_1, N_2 \geq N + \max(\mathsf{b}_F, 1) - 1$.

Set $\mu = N_1 - N + 2$; define $U(X) := (X^\mu - 1)$, $\vartheta = N/n$, and $\mathsf{z}(X) = \nu_{\mathbb{K}\setminus\mathbb{H}}(X)$;
Define $\mathsf{T}(X) := \sum_{j=1}^N = \mathbf{t}_j \lambda_j^{\mathbb{K}}(X)$;
Let $\left\{r_j^{\mathbb{K}}(X) = \frac{\lambda_j^{\mathbb{K}}(X) - \lambda_j^{\mathbb{K}}(0)}{X} U(X)\right\}_{j\in[N]}$, $\left\{r_i^{\mathbb{H}}(X) = \frac{\lambda_i^{\mathbb{H}}(X)\mathsf{z}(X) - \lambda_i^{\mathbb{H}}(0)}{X} U(X)\right\}_{i\in[n]}$,
and $\left\{Q_j(X) = \frac{(\mathsf{T}(X) - \mathbf{t}_j)\lambda_j^{\mathbb{K}}(X)}{\nu_{\mathbb{K}}(X)}\right\}_{j\in[N]}$.
Compute $\mathsf{ek}_{\mathbf{t},n} := \left[(r_j^{\mathbb{K}}(s))_{j=1}^N, (r_i^{\mathbb{H}}(s))_{i=1}^n, U(s), \nu_{\mathbb{K}}(s), s\nu_{\mathbb{K}}(s), (Q_j(s))_{j=1}^N, \mathsf{T}(s)\right]_1$;
Compute $\mathsf{vk}_{\mathbf{t},n} := [1, U(s), \mathsf{z}(s)U(s), \nu_{\mathbb{K}}(s)U(s), \mathsf{T}(s)U(s)]_2$;
Return $(\mathsf{ek}_{\mathbf{t},n}, \mathsf{vk}_{\mathbf{t},n})$.

Prove$(\mathsf{ek}_{N,n}, \mathsf{c}_\mathbf{f}, (\mathbf{f}, \rho_\mathsf{F}(X)))$**:** // $\mathsf{c}_\mathbf{f} = \left[\sum_i \mathbf{f}_i \lambda_i^{\mathbb{H}}(s) + \rho_\mathsf{F}(s)\nu_{\mathbb{H}}(s)\right]_1$, $deg(\rho_\mathsf{F}) = \mathsf{b}_F$.

Compute $\mathbf{m} = (m_1, \ldots, m_N)$ s.t. $\forall j : \mathbf{t}_j$ appears m_j times in $\mathbf{f}$; samples $\rho_m \leftarrow\$ \mathbb{F}$;
Compute $[m(s)]_1 \leftarrow \sum_{j=1}^N m_j \cdot \left[\lambda_j^{\mathbb{K}}(s)\right]_1 + \rho_m \cdot [\nu_{\mathbb{K}}(s)]_1$; // n scalar mults
Sample $R_S, \rho_S \leftarrow\$ \mathbb{F}$ and compute $[S(s)]_1 \leftarrow R_S \cdot s + \rho_S \cdot \nu_{\mathbb{K}}(s)$;
$\beta \leftarrow \mathsf{RO}(\mathsf{vk}_{N,n} \| (\mathsf{c}_\mathbf{t}, \mathsf{c}_\mathbf{f}) \| [m(s)]_1)$ //Fiat-Shamir challenge.
Sample $\rho_A \leftarrow\$ \mathbb{F}$, $\rho_B(X) \leftarrow\$ \mathbb{F}_{\leq 1}[X]$;
Let $A_j \leftarrow m_j/(\mathbf{t}_j + \beta)\ \forall j = 1, \ldots, N$ and $B_i \leftarrow 1/(\mathbf{f}_i + \beta)\ \forall i = 1, \ldots, n$;
Compute $[A(s)]_1 \leftarrow \sum_{j=1}^N A_j \left[\lambda_j^{\mathbb{K}}(s)\right]_1 + \rho_A \cdot [\nu_{\mathbb{K}}(s)]_1$;
Compute $[B(s)]_1 \leftarrow \sum_{i=1}^n B_i \left[\lambda_i^{\mathbb{H}}(s)\right]_1 + \rho_B(s) \cdot [\nu_{\mathbb{H}}(s)]_1$;
Compute $Q_B(X) \leftarrow (B(X)(\mathsf{F}(X) + \beta) - 1)/\nu_{\mathbb{H}}(X)$ and $[Q_B(s)]_1$;
$(\gamma, \eta) \leftarrow \mathsf{RO}(\beta \| [A(s), B(s), Q_B(s), S(s)]_1)$;//Fiat-Shamir challenge.
Compute $B_\gamma \leftarrow B(\gamma)$, $D(X) \leftarrow B_\gamma \cdot (\mathsf{F}(X) + \beta) - 1 - Q_B(X)\nu_{\mathbb{H}}(\gamma)$;
Compute $P(X) \leftarrow ((B(X) - B(\gamma)) + \eta D(X))/(X - \gamma)$ and $[P(s)]_1$; // KZG-proof for (6.8).
Compute $\left[R_C^*(s)\right]_1 \leftarrow \sum_{m_j \neq 0} A_j \cdot \left[r_j^{\mathbb{K}}(s)\right]_1 - \vartheta^{-1} \sum_{i=1}^n B_i \cdot \left[r_i^{\mathbb{H}}(s)\right]_1 + \eta^2 R_S \cdot [U(s)]_1$
Compute $[Q_A(s)]_1 \leftarrow \sum_{m_j \neq 0} A_j \cdot \left[Q_j(s)\right]_1 + \left[\rho_A(T(s) + \beta) - \rho_m\right]_1$;
Compute $[Q_C(s)]_1 \leftarrow \left[\rho_A + \vartheta^{-1}\rho_B(s)\right]_1$;
Compute $[Q(s)]_1 \leftarrow [Q_A(s)]_1 + \eta\,[Q_C(s)]_1 + \eta^2 \left[\rho_S\right]_1$;
Return $\pi = (\left[m(s), S(s), A(s), B(s), Q_B(s), P(s), R_C^*(s), Q(s)\right]_1, B_\gamma)$.

Verify$(\mathsf{vk}_{\mathbf{t},n}, \mathsf{c}_\mathbf{f}, \pi)$**:**

Compute $[D(s)]_1 \leftarrow B_\gamma(\mathsf{c}_\mathbf{f} + \left[\beta\right]_1) - [1]_1 - \nu_{\mathbb{H}}(\gamma)\,[Q_B(s)]_1$.
Return 1 if and only if the following holds:

(i) $e([A(s)]_1, \mathsf{c}_\mathbf{t}) \cdot e((\beta + \eta) \cdot [A(s)]_1 - [m(s)]_1 + \eta^2\,[S(s)]_1, [U(s)]_2) \cdot e(\eta/\vartheta \cdot [B(s)]_1, [\mathsf{z}(s)U(s)]_2)^{-1} \cdot e([Q(s)]_1, [\nu_{\mathbb{K}}(s)U(s)]_2)^{-1} = e(\eta \cdot \left[R_C^*(s)\right]_1, [x]_2)$,

(ii) $e([B(s)]_1 + \eta\,[D(s)]_1 - \left[B_\gamma\right]_1, [1]_2) = e([P(s)]_1, \left[s - \gamma\right]_2)$

Figure 6.1: Our zero-knowledge lookup argument cq^+.

6.5.1. THE STRAW MAN SOLUTION

An alternative approach to commit to a matrix is to one-by-one vector commit to its columns. The obvious shortcoming is that the commitment scheme is not succinct in the number of columns. Nonetheless, this approach already results in a matrix lookup argument (under the assumption that the vector commitment is linearly homomorphic). In particular, consider the lookup argument that hashes together the columns $\mathbf{t}_j$ of the table $\mathbf{T}$ and the columns $\mathbf{f}_j$ of the sub-matrix $\mathbf{F}$ using a random challenge ρ computing vectors

$$\mathbf{t}^* = \sum_j \mathbf{t}_j \rho^{j-1} \qquad\qquad \mathbf{f}^* = \sum_j \mathbf{f}_j \rho^{j-1}.$$

Notice that by Schwartz-Zippel lemma we that $\mathbf{f}^* \prec \mathbf{t}^*$ implies $\mathbf{F} \prec \mathbf{T}$ with overwhelming probability. Thus, we could run a vector lookup argument over $(\mathbf{f}^*, \mathbf{t}^*)$, thanks to the linear homomorphic property of the commitment scheme the verifier can compute commitments to $\mathbf{f}^*$ and $\mathbf{t}^*$ and verify the proof. Notice the prover time complexity is $\mathsf{poly}(n, d, n)$ thanks to the $\mathbb{F}$-linearity of the precomputation algorithm. However, the verification time is linear in the number of columns. We show in the next section how to restore succinct verification time and commitment size.

6.5.2. OUR SCHEME

In fig. 6.2 we describe our scheme $\mathsf{mtx}[\mathsf{CP}]$ that runs internally a lookup argument CP for KZG-based vector commitment scheme. The proof of the following theorem is in appendix A.4. In the description of the scheme, we let $\mathbb{K}$ (resp. $\mathbb{H}$) be a multiplicative subgroup of $\mathbb{F}$ of order $N \cdot d$ (resp. of order $n \cdot d$), we let $\omega := \omega_{n \cdot d}$ be the fixed generator for $\mathbb{H}$ and we consider the following matrices and polynomial:

1. the matrix $\mathbf{R} \in \mathbb{F}^{N \times d}$ where $R_{i,j} = i$,
2. for any k the matrix $\mathbf{C}^{(k)} \in \mathbb{F}^{k \times d}$ where $C_{i,j} = j$.
3. Let $v_{\tilde{\mathbb{H}}}(X)$ be the vanishing polynomial of $\tilde{\mathbb{H}} = \{\omega^{d \cdot i + j} : j \in [1, d-1], i \in [n]\}$.

Theorem 6.4. *The lookup argument* $\mathsf{mtx}[\mathsf{CP}]$ *defined in fig. 6.2 is knowledge-sound in the AGM and ROM under the* $(N \cdot d, N \cdot d)$*-PDL assumption and assuming that* CP *is knowledge-sound. Furthermore, the protocol is zero-knowledge assuming* CP *is zero-knowledge.*

A row-column Matrix Lookup Argument. In appendix A.4.1 we consider the rows-columns sub-matrix relation where $\mathbf{F} \prec \mathbf{T}$ if and only if there exist (multi)sets $R = \{r_1, \ldots, r_n\}$ and $C = \{c_1, \ldots, c_d\}$ with $\mathbf{F}_{i,j} = \mathbf{T}_{r_i, c_j}$, and give an *rows-columns* matrix-lookup argument system $\mathsf{mtx}^*[\mathsf{CP}]$ for such a relation. Briefly, the main difference with the scheme in this section is that we commit to an additional vector $\bar{\boldsymbol{\sigma}}^C$ which is the concatenation of the vector $(c_1, \ldots, c_d)$ for n times, prove in zero-knowledge its tensor structure, and show that $\bar{\mathbf{f}}^* = \bar{\mathbf{f}} + \rho \cdot \bar{\boldsymbol{\sigma}}^C + \rho^2 \cdot \bar{\boldsymbol{\sigma}}$ is a sub-vector of $\bar{\mathbf{t}}^*$.

Derive(srs, N, d, n):

Let $\bar{\mathbf{f}}, \bar{\mathbf{r}}_N, \bar{\mathbf{c}}_N$ and $\bar{\mathbf{c}}_n$ be vectorizations of the matrices $\mathbf{F}, \mathbf{R}, \mathbf{C}^{(N)}$ and $\mathbf{C}^{(n)}$.
Compute $\mathsf{c}_{r,N} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{r}}_N), \mathsf{c}_{c,N} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{c}}_N)$ and $\mathsf{c}_{c,n} \leftarrow \mathsf{Com}(\bar{\mathbf{c}}_n)$.
Compute $(\mathsf{ek}', \mathsf{vk}') \leftarrow \mathsf{CP.Derive}(\mathsf{srs}, Nd, nd)$.
Return $(\mathsf{ek}', \mathsf{vk}_n)$ where $\mathsf{vk}_n = (\mathsf{c}_{r,N}, \mathsf{c}_{c,N}, \mathsf{c}_{c,n}, [v_{\bar{\mathbb{H}}}(s)]_2, \mathsf{vk}')$.

Preproc(srs, $\mathbf{T}, \rho_T$):

Let $\bar{\mathbf{t}}$ be vectorization of the matrix $\mathbf{T}$.
Compute $(\mathsf{aux}_{T,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{t}}, \rho_T)$,
$(\mathsf{aux}_{R,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{r}}_N)$,
$(\mathsf{aux}_{C,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{c}}_N)$.
Let $\mathsf{aux}_i = (\mathsf{aux}_{T,di+j}, \mathsf{aux}_{R,di+j}, \mathsf{aux}_{C,di+j})_{j\in[d]}$.
Return $(\mathsf{aux}_i)_{i\in[N]}$.

Prove(ek, $(\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}), \mathbf{F}, (\mathsf{aux}_j)_{j\in K}$): //$\mathbf{T}_{|K} = \mathbf{F}$, $K = \{k_1, \ldots, k_n\}$.

Let $\mathbf{S}$ be s.t. $\mathbf{S}_{i,j} = k_i$ for $i \in [n], j \in [d]$.
Let $\sigma(X)$ be the randomized low-degree encoding over $\mathbb{H} = \langle\omega\rangle$ of the vectorization of $\mathbf{S}$.
Compute $w(X)$ such that $\sigma(\omega \cdot X) - \sigma(X) = w(X) \cdot v_{\bar{\mathbb{H}}}(X)$.
$(\rho, \zeta) \leftarrow \mathsf{RO}(\mathsf{vk}_n \| (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}) \| (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w))$. //Fiat-Shamir challenge.
Compute $z \leftarrow \sigma(\omega \cdot \zeta)$.
Compute proofs π_R and $\pi_{R'}$ for $\hat{\mathcal{R}}_{\mathsf{eval}}(\omega \cdot \zeta, z; \sigma(X)) = 1$ and $\hat{\mathcal{R}}_{\mathsf{eval}}(\zeta, z; \sigma(\omega \cdot X)) = 1$;
Let π^* proof for $\hat{\mathcal{R}}_{\mathsf{zklookup}}((N \cdot d, n \cdot d); \varepsilon; (\bar{\mathbf{t}}^*, \bar{\mathbf{f}}^*)) = 1$ where

$$\bar{\mathbf{t}}^* = \bar{\mathbf{t}} + \rho \cdot \bar{\mathbf{c}}_N + \rho^2 \cdot \bar{\mathbf{r}}_N \qquad \bar{\mathbf{f}}^* = \bar{\mathbf{f}} + \rho \cdot \bar{\mathbf{c}}_n + \rho^2 \cdot \bar{\boldsymbol{\sigma}} \tag{6.13}$$

Return $\pi = ([\sigma(s)]_1, [\sigma(\omega \cdot s)]_1, [w(s)]_1, \pi_R, \pi_{R'}, \pi^*, z)$.

Verify(vk$_n$, $(\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}), \pi$):

Parse the proof $\pi = (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w, \pi_R, \pi_{R'}, \pi^*, z)$.
$(\rho, \zeta) \leftarrow \mathsf{RO}(\mathsf{vk}_n \| (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}) \| (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w))$. //Fiat-Shamir challenge.
Compute $\mathsf{c}^*_\mathbf{T} \leftarrow \mathsf{c}_\mathbf{T} + \rho \mathsf{c}_{c,N} + \rho^2 \mathsf{c}_{r,N}$ and $\mathsf{c}^*_\mathbf{F} \leftarrow \mathsf{c}_\mathbf{F} + \rho \mathsf{c}_{c,n} + \rho^2 \mathsf{c}_{R,n}$.
Return 1 if the following checks hold (else 0):

(i) $\mathsf{Verify}_{\mathsf{eval}}(\mathsf{ck}, (\mathsf{c}_{R,n}, \omega \cdot \zeta, z)) = 1$,
(ii) $\mathsf{Verify}_{\mathsf{eval}}(\mathsf{ck}, (\mathsf{c}_{R',n}, \zeta, z)) = 1$,
(iii) $e(\mathsf{c}_{R',n} - \mathsf{c}_{R,n}, [1]_2) = e(\mathsf{c}_w, [v_{\bar{\mathsf{k}}}(s)]_2)$,
(iv) $\mathsf{CP.Verify}(\mathsf{srs}, \mathsf{vk}', (\mathsf{c}^*_\mathbf{T}, \mathsf{c}^*_\mathbf{F}), \pi^*) = 1$.

Figure 6.2: Our Matrix Lookup Argument mtx[CP].

Table 6.1: Summary of efficiency of our constructions for matrix lookups. The relation considered is parametrized with table size of size $N \times d$ and looked-up submatrix of size $n \times d$. $\mathbb{P}$ is the cost of one pairing. Proof size includes commitment to the witness.

Scheme	Preprocessing	Proof size	Time (P)	Time (V)
$\mathsf{mtx}^{\mathsf{longprf}}[\mathsf{zkcq}^+]$ (section 6.5.1)	$O(dN\log N)\mathbb{F},\mathbb{G}$	$(d+9)\mathfrak{g}_1+1\mathfrak{f}$	$O(nd)\mathbb{G}_1+O(nd\log n)\mathbb{F}$	$d\mathbb{G}_1+7\mathbb{P}$
$\mathsf{mtx}[\mathsf{zkcq}^+]$ (fig. 6.2)	$O(dN\log dN)\mathbb{F},\mathbb{G}$	$16\mathfrak{g}_1+2\mathfrak{f}$	$O(nd)\mathbb{G}_1+O(nd\log(nd))\mathbb{F}$	$13\mathbb{P}$
[23]	$O(dN\log dN)\mathbb{F},\mathbb{G}$	$20\mathfrak{g}_1+6\mathfrak{f}$	$O(nd\log nd)\mathbb{G}_1+O(nd\log(nd))\mathbb{F}$	$23\mathbb{P}$

6.5.3. CONCRETE EFFICIENCY

In table 6.1, we describe the complexity of proving a matrix lookup in a table T described by a matrix of size $N \times d$. The size of the submatrix we are looking up in the larger table is $n \times d$. In appendix A.6, we describe a breakdown of efficiency measurements for our fully zero-knowledge construction ($\mathsf{mtx}[\mathsf{zkcq}^+]$). Our naive scheme, the one derived from the observations in section 6.5.1, and the scheme in appendix A.4.1 have efficiency analyses which follow similarly. The values for [23] are taken directly from the paper, the number of pairings in verification is computed by simple inspection of the protocol, the extra $O(\log nd)$ factor in the number of exponentiations in $\mathbb{G}_1$ for the prover arises from their sub-protocol adapted from [6].

6.6. ZERO-KNOWLEDGE DECISION TREE STATISTICS

A decision tree is an algorithm that, upon an input, performs a finite sequence of adaptive queries on the input and eventually outputs a value. Concretely, we consider binary decision trees where the inputs are vectors in $[B]^d$ for natural numbers d and B, where the queries are comparisons and the outputs (often called the labels) are in $[B]$. We let N_{tot} be the number of nodes in a decision tree T, and we index the root node with 1. A binary tree with N_{tot} nodes and where each node has either zero children or exactly two children, has $N_{\mathsf{leaf}} := (N_{\mathsf{tot}}+1)/2$ leaf nodes, and the remaining $N_{\mathsf{int}} = N_{\mathsf{tot}} - N_{\mathsf{leaf}}$ nodes are called internal nodes (including the root node). We index the internal nodes of the decision tree with numbers in $[N_{\mathsf{int}}]$. The computation of a decision tree T upon input $\mathbf{x}$, denoted as $\mathsf{T}(\mathbf{x})$, consists of a traversal of the tree from the root node to a leaf. During the traversal, the computation fetches, from each internal node i, a threshold t_i and a feature index $e_i \in [d]$. If $x_{e_i} < t_i$, the computation continues recurring on the left child of node i, and otherwise, to the right child. Once reaches a leaf, the computation outputs the label v_i assigned to the leaf i as the final output.

Therefore, seen as a data structure, a decision tree T is made by a binary tree (namely, the *structure* of the tree), by the values d_i, t_i for each internal node i, and by the label v_i for each leaf node i. We refer to this encoding of a tree as the *standard encoding*. We define $\mathcal{T}_{N_{\mathsf{tot}},B,d}$ to be the set of decision trees with N_{tot} nodes that maps vector in $[B]^d$ to the co-domain $\mathbb{F}$.

Quasi-Complete Decision Tree. We define the notion of *quasi-complete* decision tree. The difference with a standard tree is that during the traversal, the computation fetches from each internal node i two vectors $\mathbf{E}_i$ and $\mathbf{T}_i$, we call the vector $\mathbf{E}_i \in \{0,1\}^d$ the feature vector associated to the node i and vector $\mathbf{T}_i \in [B]^d$ the threshold vector associated to the node i. The computation continues recurring on the left child of node i if $\forall j \in [d]$:

$\mathbf{E}_{i,j} = 1 \Rightarrow x_j < \mathbf{T}_{i,j}$, on to the right child of the node i if $\forall j \in [d] : \mathbf{E}_{i,j} = 1 \Rightarrow x_j \geq \mathbf{T}_{i,j}$, or outputs $\perp$ if neither of the two conditions holds. The pseudo-code of the evaluation of a quasi-complete decision tree is in fig. A.4 in appendix A.5.

Similarly to decision trees, we define $\mathcal{T}^*_{N_{\mathsf{tot}},B,d}$ to be the set of quasi-complete decision trees with N_{tot} nodes that maps feature vector in $[B]^d$ to the co-domain $\mathbb{F}$. Notice that when for any node j the (row) vector $\mathbf{E}_j$ is an elementary vector (namely with only one position set to 1) then the quasi-complete decision tree is indeed a standard decision tree thus $\mathcal{T}_{N_{\mathsf{tot}},B,d} \subset \mathcal{T}^*_{N_{\mathsf{tot}},B,d}$.

The class of quasi-complete decision trees defines a correct but not complete computational model. In fact, every input is either correctly labelled to one label or to the error message $\perp$. Being a more general class of computation than standard decision trees, it is easier to decide whether a data structure is a quasi-complete decision tree than to decide if it is a standard decision tree. This allows for faster prover time. On the other hand, an adversary that commits to (strictly) quasi-complete decision tree (namely, a decision tree in $\mathcal{T}^*_{N_{\mathsf{tot}},B,d} \setminus \mathcal{T}_{N_{\mathsf{tot}},B,d}$) cannot prove contradicting statements, in particular, we require that it cannot prove any statistics on an input $\mathbf{x}$ whenever $\mathsf{T}(\mathbf{x}) = \perp$.

6.6.1. SECURITY MODEL

We consider the scenario where a *model producer* commits to a decision tree T, the model producer can delegate the computation of statistics on a set of data points and predictions over T to a server, a user can obtain such statistics. Informally, we require integrity of the computation, namely the statistics are correctly computed over the set of data points and predictions over the committed decision tree T, and privacy, namely the user does not learn anything more than the validity of such statistics.

We consider an adversarial model where either the model producer and the server can be corrupted, or the user is corrupted. Previous work considered only the case where the model producer is honest [11] (and either the server or user are corrupted). Notice that a corrupted model producer could commit to a *useless/bogus* decision tree. Unfortunately, we cannot do anything to prevent that. On the other hand, we would like to prevent the corrupted model producer and corrupted server can convince the user of the validity of *incoherent* statistics. For example, an attacker should not be able to convince the user that simultaneously $\mathsf{T}(\mathbf{x}) = 1$ and $\mathsf{T}(\mathbf{x}) = 0$ for a data point $\mathbf{x}$.

To formalize such property, we use the notion of knowledge soundness for argument systems. In particular, we require that whenever the verifier is convinced (w.r.t. a commitment c) of the statistic over a set of data points, there must exist an extractor that outputs an opening of the commitment to a decision tree T where such a statistic over such data is correct. Notice the commitment to the decision tree is binding. Thus we must obtain coherent statistics over many queries on the same committed decision tree. To optimize the efficiency of the statistic evaluations, we split in two parts the generation of a valid commitment from the evaluation of a proof for a given tuple statistic/data points.

Definition 6.6. *Let $\mathcal{S}$ be an arbitrary set of tuples (S, m) such that $S : [B]^m \to \{0,1\}^*$ and $m \in \mathbb{N}$ where S is an efficiently computable function (a statistic). A (commit-and-prove)* decision-tree-statistic argument *for a set of statistics $\mathcal{S}$ is a tuple* $\mathsf{zkDT} =$

(KGen, Com, VerCom, Derive, Prove, Verify) *where:*

(i) $\mathsf{CS}_{DT} = (\mathsf{KGen}, \mathsf{Com}, \mathsf{VerCom})$ *define an extractable commitment scheme for the domain* $\mathcal{T}^*$ *of (quasi-complete) decision tree. In particular,* KGen *takes in input a natural number* N_{tot} *the maximum number of nodes, and the natural numbers* B *and* d*, besides the security parameter and generates a commitment key for the set* $\mathcal{T}^*_{B,d,N_{\mathsf{tot}}}$.

(ii) $\mathsf{CP}_{DT} = (\mathsf{Derive}, \mathsf{Prove}, \mathsf{Verify})$ *define a Universal CP-SNARK for the indexed CP-relation* $\hat{\mathcal{R}}_{\mathsf{DTstat}}$ *defined below.*

$$\hat{\mathcal{R}}_{\mathsf{DTstat}} = \left\{ \mathsf{pp}; (S, m); y, (\mathbf{x}_j)_{j\in[m]}); \mathsf{T} : \begin{array}{c} y = S(\mathsf{T}(\mathbf{x}_1), \ldots, \mathsf{T}(\mathbf{x}_m)), \\ \forall i : \mathsf{T}(\mathbf{x}_i) \neq \bot, \ (S, m) \in \mathcal{S} \end{array} \right\}.$$

6.6.2. THE EXTENDED ENCODING OF DECISION TREES

We introduce an alternative encoding of a decision tree as a data structure before presenting our zero-knowledge decision-tree statistics argument. We follow the work of Chen *et al.* [15]. In particular, we define a d-dimensional *box* as a tuple of vectors in $[B+1]^d$, where the first vector defines the *left bounds* and the second vector defines the *right bounds*. We say that a vector $\mathbf{x} \in [B]^d$ *is contained* in a box $(\mathbf{b}^-, \mathbf{b}^-)$ if $\mathbf{b}^- \leq \mathbf{x} < \mathbf{b}^-$. We can assign to each node of a decision tree a $d-$dimensional box. In particular, we denote with $(\mathbf{N}^-_i, \mathbf{N}^-_i)$ the box assigned to the i-th node in the tree and with $\mathbf{N}^-, \mathbf{N}^-$ the tuple of matrices of all the boxes of a decision tree (mapping the i-th row to the box of i-th node).

We can associate a (quasi-complete) decision tree to a tuple of matrices, below we define such a relation:

Definition 6.7. *Given a quasi-complete decision tree* T *with* N_{tot} *nodes and given matrices* $\mathbf{N}^-, \mathbf{N}^-$*, we say that* $(\mathbf{N}^-, \mathbf{N}^-)$ *is a* boxes-encoding *of* T *if*

1. $\mathbf{N}^-_1 = \mathbf{0}$ *and* $\mathbf{N}^-_1 = \mathbf{B} + \mathbf{1}$*, where* $\mathbf{0}$ *(resp.* $\mathbf{1}$ *and* $\mathbf{B}$*) is the vector of all 0 (resp. of all* 1 *and of all* B*).*

2. *Let* $p \in [N_{\mathsf{int}}]$ *be the index of a node and let* l *and* r *respectively be the indexes of the left child and right child of the node with index* p*.*

$$\mathbf{N}^-_l - \mathbf{N}^-_p = \mathbf{0} \qquad \mathbf{N}^-_r - \mathbf{N}^-_p = \mathbf{0} \tag{6.14}$$

$$\mathbf{E}_p \circ (\mathbf{N}^-_l - \mathbf{T}_p) = \mathbf{0} \qquad \mathbf{E}_p \circ (\mathbf{N}^-_r - \mathbf{T}_p) = \mathbf{0} \tag{6.15}$$

$$(\mathbf{1} - \mathbf{E}_p) \circ (\mathbf{N}^-_l - \mathbf{N}^-_p) = \mathbf{0} \qquad (1 - \mathbf{E}_p) \circ (\mathbf{N}^-_r - \mathbf{N}^-_p) = \mathbf{0} \tag{6.16}$$

The computation, through a boxes-encoding, of a decision tree $\mathsf{T}(\mathbf{x})$ consists in finding the index k of the leaf whose box contains $\mathbf{x}$ and outputs the label associated with such a leaf. For a quasi-complete decision tree, such an index k might not exist. We formalize this in the next definition and prove such a computational equivalence in the next lemma whose proof is in appendix A.5.

Definition 6.8. *Let* T *be a quasi-complete decision tree with* N_{tot} *nodes (with domain* $[B]^d$*) and* $(\mathbf{N}^-, \mathbf{N}^-)$ *be a boxes-encoding of* T*. For any* $\mathbf{x} \in [B]^d$*, if* $\mathbf{x}$ *is contained in the box of a leaf of* T *define the index of the leaf as* $k_{\mathsf{T}}(\mathbf{x})$ *such that* $\mathbf{x}$ *is contained in* $(\mathbf{N}^-_{k_{\mathsf{T}}(\mathbf{x})}, \mathbf{N}^-_{k_{\mathsf{T}}(\mathbf{x})})$ *else* $k_{\mathsf{T}}(\mathbf{x})$ *is set to* $\perp$*.*

Whenever it is clear from the context, we will omit the subscript T and write $k(\mathbf{x})$ to refer to such an index.

Lemma 6.5. *Let* T *be a quasi-complete decision tree with* N_{tot} *nodes and* $(\mathbf{N}^-, \mathbf{N}^-)$ *be a boxes-encoding of* T*. Let* $\mathbf{v}$ *be the vector of the labels assigned to the leaf nodes of* T*, namely for any* $i \in [N_{\mathsf{int}}+1, N_{\mathsf{tot}}]$*, we have* v_i *as the label assigned to the* i*-th leaf. For any* $\mathbf{x} \in [B]^d$*,* $\mathsf{T}(\mathbf{x}) = v_{k(\mathbf{x})}$ *or* $\mathsf{T}(\mathbf{x}) = \perp$*.*

As corollary of the above lemma, we have that the boxes of leaf do not overlap because no vector $\mathbf{x}$ can be contained in more than one of the boxes of the leaves.

Before giving the next definition, we set some notation: given a decision tree, we say that node p splits at coordinate $i^* \in [d]$ if i^* is a coordinate where p's left and right child boundaries are different, namely, $\mathbf{N}^-_{p,i} \neq \mathbf{N}^-_{\ell,i}$ and $\mathbf{N}^-_{p,i} \neq \mathbf{N}^-_{r,i}$ where ℓ and r are the left and right child of p. We are ready to describe our (more redundant but ZKP-friendly) encoding of a quasi-complete decision tree.

Definition 6.9. *Let* T *be a quasi-complete decision tree with* N_{tot} *nodes. Let* $\mathcal{T} = (\mathbf{N}^-, \mathbf{N}^-, \mathbf{v}, \mathbf{L}, \mathbf{R}, \mathbf{E})$ *be a tuple of matrices (described below). We say that* $\mathcal{T}$ *is an extended encoding of* T *if the following conditions hold:*

(i) $(\mathbf{N}^-, \mathbf{N}^-)$ *is a boxes-encoding of* T*;*

(ii) $\mathbf{v}$ *is the vector of the labels assigned to the leaf nodes of* T*;*

(iii) $\mathbf{L}$ *(resp.* $\mathbf{R}$*) is the* $N_{\mathsf{int}} \times N_{\mathsf{tot}}$ *bit matrix whose* p*-th row is the elementary vector* $\mathbf{e}^{\mathsf{T}}_{\ell}$ *(resp.* $\mathbf{e}^{\mathsf{T}}_{r}$*) if* ℓ *is the left (resp.* r *is the right) child of node* p*'s in* T*,*

(iv) $\mathbf{E} \in \{0,1\}^{N_{\mathsf{int}} \times d}$ *is the bit matrix such that its* p*-th row and* i *column is* 1 *iff the node* p *splits at coordinate* i*.*

Let Encode *be the algorithm that, given a quasi-complete decision tree* T*, computes the extended encoding of* T*.*

Let the matrices $\mathbf{P}^-, \mathbf{P}^- \in \mathbb{F}^{N_{\mathsf{int}} \times d}$ describe the boxing encodings of the internal nodes, and $\mathbf{F}^-, \mathbf{F}^- \in \mathbb{F}^{N_{\mathsf{leaf}} \times d}$ describe the boxing encodings of the leaves. Thus:

$$\mathbf{N}^- = \begin{pmatrix} \mathbf{P}^- \\ \| \\ \mathbf{F}^- \end{pmatrix} \text{ and } \mathbf{N}^- = \begin{pmatrix} \mathbf{P}^- \\ \| \\ \mathbf{F}^- \end{pmatrix}.$$

The function Encode in definition 6.9 is injective but not surjective. In the next lemma (whose proof is in appendix A.5), we give sufficient conditions for belonging in the image of Encode.

Lemma 6.6. *Consider a tuple* $(\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$ *such that the following constraints hold:*

a) The following equations hold:

$$\mathbf{N}_1^- = \mathbf{0}, \mathbf{N}_1^+ = \mathbf{B} + \mathbf{1}, \tag{6.17}$$
$$\mathbf{L} \cdot \mathbf{N}^- = \mathbf{P}^-, \mathbf{R} \cdot \mathbf{N}^+ = \mathbf{P}^-, \tag{6.18}$$
$$\mathbf{E} \circ (\mathbf{L} \cdot \mathbf{N}^- - \mathbf{R} \cdot \mathbf{N}^-) = \mathbf{0} \tag{6.19}$$
$$(\mathbf{1} - \mathbf{E}) \circ (\mathbf{P}^- - \mathbf{R} \cdot \mathbf{N}^-) = \mathbf{0}, \qquad (\mathbf{1} - \mathbf{E}) \circ (\mathbf{P}^- - \mathbf{L} \cdot \mathbf{N}^-) = \mathbf{0} \tag{6.20}$$

b) All the boxes are not empty. Namely, for all i, j we have $\mathbf{N}^-_{i,j} < \mathbf{N}^-_{i,j}$.

c) The matrix $\begin{pmatrix} \mathbf{L} \\ \mathbf{R} \end{pmatrix}$ is a (row) permutation of the (squared) matrix $(\mathbf{0} \| \mathbf{I}_{N_{\mathsf{tot}}-1})$ (the matrix whose rows are the row vectors $(\mathbf{e}_i)_{i \in [2, N_{\mathsf{tot}}]}$ of length N_{tot}).

Then there exists a quasi-complete decision tree T with N_{tot} nodes such that $\mathsf{Encode}(\mathsf{T}) = (\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$.

6.6.3. EXTRACTABLE COMMITMENT TO DECISION TREES

In a nutshell our commitment procedure on input a decision tree computes the encoding described in section 6.6.2, then it commits to the matrices $\mathbf{F}^-, \mathbf{F}^-$ and $\mathbf{v}$ and prove in zero-knowledge the constraints from theorem 6.6. We can implement the latter zero-knowledge proof using a general-purpose R1CS circuit describing the constraints of the lemma, however, the size of the circuit would be $O(dN^2_{\mathsf{tot}})$, in fact, we would need to commit to the remaining matrices $\mathbf{P}^-, \mathbf{P}^-, \mathbf{L}, \mathbf{R}$ and $\mathbf{E}$ and we would need already $O(dN^2_{\mathsf{tot}})$ multiplication gates for eq. (6.18). We show how to remove the quadratic dependency from the number of total nodes. The main idea is to notice that $\mathbf{L}$ and $\mathbf{R}$ have sparsity linear in N_{tot}, thus we can use techniques from [7] to commit to such sparse matrices and then prove in zero-knowledge that the constraints in item c) of theorem 6.6 hold for the committed matrices. The remaining constraints can be proved in $O(dN_{\mathsf{tot}} \log(dN_{\mathsf{tot}}))$.

The building blocks. Consider the following (indexed) CP-relations:

$$\hat{\mathcal{R}}_{\mathsf{lin}} = \{\mathsf{pp}; \varepsilon; (\mathbf{M}, \mathbf{N}, \mathbf{R}) : \mathbf{M} \cdot \mathbf{N} = \mathbf{R}\} \tag{6.21}$$
$$\hat{\mathcal{R}}_{\mathsf{had}} = \{\mathsf{pp}; \varepsilon; (\mathbf{M}, \mathbf{N}) : \mathbf{M} \circ \mathbf{N} = \mathbf{0}\} \tag{6.22}$$
$$\hat{\mathcal{R}}_{\mathsf{perm}} = \left\{\mathsf{pp}; (N, i(X)); \varepsilon; p(X) : \exists \pi, \forall j \in [N] : i(\pi(\omega^j)) = p(\omega^j)\right\} \tag{6.23}$$
$$\hat{\mathcal{R}}_{\mathsf{shift}} = \{\mathsf{pp}; S; \varepsilon; (\mathbf{v}, \mathbf{u}) : \mathbf{v}_i = \mathbf{u}_{(i+S \pmod{|\mathbf{u}|})}\} \tag{6.24}$$
$$\hat{\mathcal{R}}_{\mathsf{range}} = \left\{\mathsf{pp}; (B, n, d); \varepsilon; \mathbf{X} : \mathbf{X} \in [B]^{n \times d}\right\} \tag{6.25}$$
$$\hat{\mathcal{R}}_{\mathsf{sm}} = \{\mathsf{pp} : K; \varepsilon; \mathbf{M} : \mathbf{M}_{|K} = \mathbf{0}\} \tag{6.26}$$

Our scheme uses CP-SNARKs for all the relations above as building blocks. The first three relations are standard, and CP-SNARKs for them can be found in the related work. Given a CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{lin}}$, we can define a CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{shift}}$ in fact that the shifting operator can be described through a linear transformation. The latter linear transformation can be public, thus the underlying CP-SNARK (for $\hat{\mathcal{R}}_{\mathsf{lin}}$) does not need to be zero-knowledge w.r.t. the first matrix $\mathbf{M}$, in particular, a commitment to such a matrix

could be part of the index polynomials. A CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{range}}$ can be realized using our lookup argument and considering the table $\mathbf{b} = (j)_{j\in[B]}$ and proving that the vectorization $\bar{\mathbf{x}}$ of $\mathbf{X}$ is such that $\bar{\mathbf{x}} \prec \mathbf{b}$. Finally, a CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{sm}}$ can be easily realized by committing to a matrix $\bar{\mathbf{T}}$ such that $\bar{\mathbf{T}}_K = \mathbf{T}$ and 0 everywhere else and to the vanishing polynomial in ν_K in $\mathbb{G}_2$ as part of the index. At proving time, the prover returns as proof a commitment to the quotient polynomial q such that $f'(X) = q(X)\cdot\nu_K(X)$ where $f'(X)$ is the polynomial associated to the matrix $\mathbf{M} - \bar{T}$. At verification time the verifier checks $e(\mathsf{c}_{\mathbf{M}} - \mathsf{c}_{\bar{\mathbf{T}}}, [1]_2) = e(\pi, [\nu_K]_2)$.

For the CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{lin}}$, we require two different commitment schemes, one for the first matrix and one for the other two. In particular, we consider an alternative way to commit to matrices following the work of [7, 20]. Let $\mathbf{M}$ be a *basic matrix*, namely a matrix whose rows are elementary vectors. Let $\mathbb{H}$ be any fixed subgroup with $|\mathbb{H}| \geq N_{\mathsf{tot}}$ [8] of $\mathbb{F}$ with generator ω. For any basic matrix $\mathbf{M} \in \{0,1\}^{n\times k}$ and $n, k \in \mathbb{N}$, let $col_{\mathbf{M}}(X)$ be the (low-degree) polynomial such that $col_{\mathbf{M}}(\omega^i) = \omega^j$ where the i-th row of $\mathbf{M}$ is the vector $\mathbf{e}_j^\top$ (notice that $col_{\mathbf{M}}$ is the LDE of the vector whose i-th element is the value ω^j). We define the sparse (hiding) commitment of a matrix $\mathbf{M}$ as a (hiding) polynomial commitment of $col_{\mathbf{M}}$. Namely, we define:

$$\mathsf{sparseCom}(\mathsf{ck}, \mathbf{M}, \rho) := \mathsf{Com}(\mathsf{ck}, col_{\mathbf{M}}, \rho).$$

6

Notice that, by the above definition, a sparse commitment to a basic matrix $\mathbf{M}$ has a dual interpretation (as a sparse matrix or as a vector col).

Let $\mathsf{CP}_{\mathsf{lin}}$ be a CP-SNARK for the $\hat{\mathcal{R}}_{\mathsf{lin}}$ relation where the first matrix is committed using sparseCom while the other matrices are committed with the matrix commitment scheme from section 6.2.3. An instantiation of such a scheme can be found for the matrix-times-vector case (namely, $\mathbf{N} \in \mathbb{F}^{n\times 1}$) in Baloo by [7] (see Sections 5.2, 5.3 and 5.4 of the paper). We show a generalization to matrix-times-matrix case in appendix A.5.4. We write $\boxed{\mathbf{M}}$ to underline that the matrix $\mathbf{M}$ is committed with a sparse matrix commitment. For example, we can write $(\mathsf{pp}, \varepsilon; \boxed{\mathbf{M}}, \mathbf{N}, \mathbf{R}) \in \hat{\mathcal{R}}_{\mathsf{lin}}$ to identify the statement that there are commitments $\mathsf{c}_M, \mathsf{c}_N, \mathsf{c}_R$ where the first is a sparse matrix commitment and that open to $\mathbf{M}, \mathbf{N}$ and $\mathbf{R}$ with $\mathbf{M}\cdot\mathbf{N} = \mathbf{R}$.

Let $\mathsf{CP}_{\mathsf{had}}$ be a CP-SNARK for the $\hat{\mathcal{R}}_{\mathsf{had}}$ relation where all the matrices are committed using the commitment scheme from section 6.2.3. Notice that a CP-SNARK for our matrix commitment scheme for such a CP-relation derives directly from CP-SNARK for vector commitment. Finally, let $\mathsf{CP}_{\mathsf{perm}}$ be a CP-SNARK for the CP-relation $\hat{\mathcal{R}}_{\mathsf{perm}}$. The permutation argument of Plonk [2] is a CP-SNARK for such a relation.

The Extractable Commitment to Decision Tree. We define our extractable commitment scheme for the domain of quasi-complete decision trees. The main idea is, as part of the proof of opening, to commit to the matrices $\mathbf{L}$ and $\mathbf{R}$ through sparse commitments to basic matrices and then prove the linear relations from theorem 6.6 in zero-knowledge with a complexity that is linear in the sparsity of the matrices and the dimension d. The additional constraints on the two matrices $\mathbf{L}$ and $\mathbf{R}$ are proved using the permutation argument. To improve readability, we list below shortcuts used in the protocol's descrip-

[8] Alternatively, we can consider the same subgroup used for the matrix commitment and thus $|\mathbb{H}| = N_{\mathsf{tot}} \cdot d$.

KGen$(1^n, (N_{\mathsf{tot}}, B, d))$**:**

Sample a type-3 pairing group pp_G with security level n.
Set $\mathsf{ck}' \leftarrow (\mathsf{pp}_G, ([s^i]_1)_{i\in[N_1]}, ([s^i]_2)_{i\in[N_2]}$ for random secrets $s \leftarrow\$ \mathbb{Z}_q$.
Compute $\mathsf{srs}_{\mathsf{sm},1} \leftarrow \mathsf{CP}_{\mathsf{sm}}.\mathsf{Derive}(\mathsf{ck}', [N_{\mathsf{int}}])$, $\mathsf{srs}_{\mathsf{sm},2} \leftarrow \mathsf{CP}_{\mathsf{sm}}.\mathsf{Derive}(\mathsf{ck}', (N_{\mathsf{int}}, N_{\mathsf{tot}}])$ and $\mathsf{srs}_{\mathsf{sm},3} \leftarrow \mathsf{CP}_{\mathsf{sm}}.\mathsf{Derive}(\mathsf{ck}', \{1\})$.
Compute $\mathsf{srs}_{\mathsf{perm}} \leftarrow \mathsf{CP}_{\mathsf{perm}}.\mathsf{Derive}(\mathsf{ck}', (N_{\mathsf{tot}} - 1, id))$.
Compute $\mathsf{srs}_{\mathsf{range}} \leftarrow \mathsf{CP}_{\mathsf{range}}.\mathsf{Derive}(\mathsf{ck}', (B, N_{\mathsf{tot}}, d))$.
Compute $\mathsf{srs}_{\mathsf{shift}} \leftarrow \mathsf{CP}_{\mathsf{shift}}.\mathsf{Derive}(\mathsf{ck}', N_{\mathsf{int}})$.
Return $\mathsf{ck} := (\mathsf{ck}', [b(s)]_1, \mathsf{srs}_{\mathsf{perm}}, \mathsf{srs}_{\mathsf{range}}, \mathsf{srs}_{\mathsf{shift}}, (\mathsf{srs}_{\mathsf{sm},j})_{j\in[3]})$.

Com$(\mathsf{ck}, \mathsf{T}, \rho_{\mathsf{T}})$**:**

Compute $(\mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{N}^-, \mathbf{N}^-, \mathbf{v}) \leftarrow \mathsf{Encode}(\mathsf{T})$, parses ρ_{T} as $(\rho_v, \rho_{\_}, \rho_{\_})$.
$\mathsf{c}_v \leftarrow \mathsf{Com}(\mathsf{ck}, \mathbf{v}, \rho_v)$. // Parse $\mathbf{v}$ as a $N_{\mathsf{tot}} \times d$ matrix whose last $d-1$ columns are empty.
$\mathsf{c}_{\_} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{F}}_{\_}, \rho_{\_})$, $\mathsf{c}_{\_} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{F}}_{\_}, \rho_{\_})$, $\mathsf{c}'_{\_} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{P}}_{\_}, \rho_{\_})$, $\mathsf{c}'_{\_} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{P}}_{\_}, \rho_{\_})$.
$\mathsf{c}_{ln} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{L}} \cdot \mathbf{N}^-,)$, $\mathsf{c}_{rn} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{R}} \cdot \mathbf{N}^-)$ and $\mathsf{c}_E \leftarrow \mathsf{Com}(\mathsf{ck}, \mathbf{E})$.
$\mathsf{c}_L \leftarrow \mathsf{sparseCom}(\mathsf{ck}, \bar{\mathbf{L}})$, $\mathsf{c}_R \leftarrow \mathsf{sparseCom}(\mathsf{ck}, \bar{\mathbf{R}})$ and $\mathsf{c}'_R \leftarrow \mathsf{sparseCom}(\underline{\mathbf{R}})$.
Let $col_{\bar{\mathbf{L}}}, col_{\bar{\mathbf{R}}}$ and $col_{\underline{\mathbf{R}}}$ be the underlying polynomials.
Prove the following statements, let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{16})$ be the proofs.

$$\pi_1, \ldots, \pi_4 : \quad (\boxed{\bar{\mathbf{L}}}, \mathbf{N}^-, \bar{\mathbf{P}}_{\_}),\ (\boxed{\bar{\mathbf{L}}}, \mathbf{N}^-, \bar{\mathbf{L}} \cdot \mathbf{N}^-),\ (\boxed{\bar{\mathbf{R}}}, \mathbf{N}^-, \bar{\mathbf{P}}_{\_}),\ (\boxed{\bar{\mathbf{R}}}, \mathbf{N}^-, \bar{\mathbf{R}} \cdot \mathbf{N}^-) \in \hat{\mathcal{R}}_{\mathsf{lin}},$$

$$\pi_5, \pi_6, \pi_7 : \quad (\bar{\mathbf{E}}, \bar{\mathbf{L}} \cdot \mathbf{N}^- - \bar{\mathbf{R}} \cdot \mathbf{N}^-),\ (\mathbf{1} - \bar{\mathbf{E}}, \mathbf{P}^- - \bar{\mathbf{R}} \cdot \mathbf{N}^-),\ (\mathbf{1} - \bar{\mathbf{E}}, \mathbf{P}^- - \bar{\mathbf{L}} \cdot \mathbf{N}^-) \in \hat{\mathcal{R}}_{\mathsf{had}},$$

$$\pi_8, \pi_9 : \quad ((B, N_{\mathsf{tot}}, d); \mathbf{N}^- - \mathbf{N}^- - \mathbf{1}) \in \hat{\mathcal{R}}_{\mathsf{range}},\quad (N_{\mathsf{tot}} - 1, id; col_{\bar{\mathbf{L}}}(X) + col_{\underline{\mathbf{R}}}(X)) \in \hat{\mathcal{R}}_{\mathsf{perm}},$$

$$\pi_{10} : \quad (N_{\mathsf{int}}, col_{\bar{\mathbf{R}}}, col_{\underline{\mathbf{R}}}) \in \hat{\mathcal{R}}_{\mathsf{shift}},$$

$$\pi_{11}, \ldots, \pi_{16} : \quad (\mathcal{N}_1; \bar{\mathbf{F}}_{\_}), (\mathcal{N}_1; \bar{\mathbf{F}}_{\_}), (\mathcal{N}_2; \bar{\mathbf{P}}_{\_}), (\mathcal{N}_2; \bar{\mathbf{P}}_{\_}), (\mathcal{N}_3; \bar{\mathbf{P}}_{\_}), (\mathcal{N}_3; \bar{\mathbf{P}}_{\_} - \mathbf{B}) \in \hat{\mathcal{R}}_{\mathsf{sm}}.$$

Return $(\mathsf{c}_{\_}, \mathsf{c}_{\_}, \mathsf{c}_v), \pi$ where $\pi = \left(\mathsf{c}'_{\_}, \mathsf{c}'_{\_}, \mathsf{c}_{ln}, \mathsf{c}_{rn}, \mathsf{c}_E, \mathsf{c}_L, \mathsf{c}_R, \mathsf{c}'_R, \boldsymbol{\pi}\right)$.

Verify$(\mathsf{ck}, \mathsf{c}_{\mathsf{T}})$**:**

Let $\mathsf{c}_{\mathsf{T}} = (\mathsf{c}_{\_}, \mathsf{c}_{\_}, \mathsf{c}_v, \pi)$ and parse π. Let $\mathsf{c}_{N,\_} \leftarrow \mathsf{c}_{\_} + \mathsf{c}'_{\_}$ and $\mathsf{c}_{N,\_} \leftarrow \mathsf{c}_{\_} + \mathsf{c}'_{\_}$.

1. Verify $\pi_1, \pi_2, \pi_3, \pi_4$ w.r.t. $(\mathsf{c}_L, \mathsf{c}_{N,\_}, \mathsf{c}'_{\_}), (\mathsf{c}_L, \mathsf{c}_{N,\_}, \mathsf{c}_{ln}), (\mathsf{c}_R, \mathsf{c}_{N,\_}, \mathsf{c}'_{\_}), (\mathsf{c}_R, \mathsf{c}_{N,\_}, \mathsf{c}_{rn})$.
2. Verify π_5, π_6, π_7 w.r.t. $(\mathsf{c}_E, \mathsf{c}_{ln} - \mathsf{c}_{rn}), ([1]_1 - \mathsf{c}_E, \mathsf{c}'_{\_} - \mathsf{c}_{rn}), ([1]_1 - \mathsf{c}_E, \mathsf{c}'_{\_} - \mathsf{c}_{ln})$.
3. Verify π_8, π_9 w.r.t. $((B, N_{\mathsf{tot}}, d); \mathsf{c}_{N,\_} - \mathsf{c}_{N,\_} - [1]_1)$ and $(N_{\mathsf{tot}} - 1, id; \mathsf{c}_L + \mathsf{c}'_R)$.
4. Verify π_{10} w.r.t. $(N_{\mathsf{int}}; (\mathsf{c}_R, \mathsf{c}'_R))$.
5. Verify $\pi_{11}, \ldots, \pi_{16}$ w.r.t. $([N_{\mathsf{int}}], \mathsf{c}_{\_}), ([N_{\mathsf{int}}], \mathsf{c}_{\_}), ((N_{\mathsf{int}}, N_{\mathsf{tot}}], \mathsf{c}'_{\_}), ((N_{\mathsf{int}}, N_{\mathsf{tot}}], \mathsf{c}'_{\_}), (\{1\}, \mathsf{c}'_{\_}), (\{1\}, \mathsf{c}'_{\_} - [b(s)]_1)$.

Figure 6.3: Our extractable commitment CS_{DT}. The value $N_1 \geq N_{\mathsf{tot}} \cdot d$, N_1 and N_2 are big enough to support all the building-block.

tion.

$$\bar{\mathbf{F}}_{-} := \begin{pmatrix} \mathbf{0} \\ \hline \mathbf{F}^{-} \end{pmatrix}, \bar{\mathbf{F}}_{-} := \begin{pmatrix} \mathbf{0} \\ \hline \mathbf{F}^{-} \end{pmatrix}, \bar{\mathbf{P}}_{-} := \begin{pmatrix} \mathbf{P}^{-} \\ \hline \mathbf{0} \end{pmatrix}, \bar{\mathbf{P}}_{-} := \begin{pmatrix} \mathbf{P}^{-} \\ \hline \mathbf{0} \end{pmatrix},$$
$$\bar{\mathbf{L}} := \begin{pmatrix} \mathbf{L} \\ \hline \mathbf{0} \end{pmatrix}, \bar{\mathbf{R}} := \begin{pmatrix} \mathbf{R} \\ \hline \mathbf{0} \end{pmatrix}, \underline{\mathbf{R}} := \begin{pmatrix} \mathbf{0} \\ \hline \mathbf{R} \end{pmatrix}, \bar{\mathbf{E}} := \begin{pmatrix} \mathbf{E} \\ \hline \mathbf{0} \end{pmatrix}$$

The padding for the matrices make them all to have N_{tot} rows. Moreover, we let $\mathbf{B}$ be the matrix whose first row is the vector $(B+1,\ldots,B+1)$ and the remaining rows are set to $\mathbf{0}$, and we let $b(X)$ be the LDE of the vectorization of such a matrix. This polynomial can be computed in $O(d\log d)$ operations, however, for simplicity, we commit to the polynomial at key-generation phase. We let id be the low-degree polynomial that evaluates $id(\omega^i) = \omega^{i+1}$ for $i \in [N_{\mathsf{tot}} - 1]$ (equivalently, the commitment $[id(s)]_1$ is a sparse-matrix commitment to the matrix $(\mathbf{0} \| \mathbf{I}_{N_{\mathsf{tot}}-1})$).

Theorem 6.7. *The commitment scheme CS_{DT} defined in Figure 6.3 is hiding, and it is an extractable commitment scheme for the domain $\{\mathcal{T}^*_{N_{\mathsf{tot}},B,d}\}_{N_{\mathsf{tot}},d,B}$ in the AGM and assuming the building blocks are knowledge-sound and zero-knowledge.*

Efficiency. The extractable commitment in this section has constant proof size when the CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{lin}}$ is instantiated with the building block described in appendix A.5.4. Its proving time is $O(dN_{\mathsf{tot}}\log(dN_{\mathsf{tot}}))$ when applied to a decision tree with d features and N_{tot} nodes. Notice that N_{tot} is usually at least one order of magnitude larger than d.

6

6.6.4. CP-SNARK FOR STATISTICS ON DECISION TREES

Consider the scheme CP_{DT} in fig. 6.4 based on the following building blocks:

1. Let $\mathsf{CP}_{\mathsf{lookup}*}$ be a CP-SNARK for the indexed CP-relation:

$$\hat{\mathcal{R}}_{\mathsf{lookup}*} = \left\{ \mathsf{pp}; (N,d,n); \varepsilon; (\mathbf{T}_j)_{j\in[m]}, (\mathbf{F}_j)_{j\in[m]} : \begin{array}{c} (\mathbf{F}_1 \| \ldots \| \mathbf{F}_m) \prec (\mathbf{T}_1 \| \ldots \| \mathbf{T}_m) \\ \forall j : |\mathbf{T}_j| = N \times d, |\mathbf{F}_j| = n \times d \end{array} \right\}$$

2. Let $\mathsf{CP}_{\mathsf{range}}$ be a CP-SNARK for the indexed CP-relation $\hat{\mathcal{R}}_{\mathsf{range}}$ in eq. (6.25).

3. Let $\mathsf{CP}_{\mathsf{stat}}$ be a CP-SNARK for the following indexed CP-relation:

$$\hat{\mathcal{R}}_{\mathsf{stat}} = \{\mathsf{pp}, (S,m); y; \mathbf{v} : S(\mathbf{v}) = y \wedge |\mathbf{v}| = m\}$$

Notice, we can easily define a CP-SNARK for $\hat{\mathcal{R}}_{\mathsf{lookup}*}$ on top of our compiler from section 6.5. Namely, we batch together the matrices $\mathbf{T}_j$ and the matrices $\mathbf{F}_j$ using a random challenge, as described in section 6.5.1, and then we run our matrix lookup argument. As corollary of theorem 6.7 and the theorem below, we have that the CP_{DT} and the commitment scheme CS_{DT} from the previous section define a decision-tree statistic argument.

Theorem 6.8. $\mathsf{CP}_{DT} = (\mathsf{Derive}, \mathsf{Prove}, \mathsf{Verify})$ *in fig. 6.4 defines an Universal CP-SNARK for the indexed CP-relation $\hat{\mathcal{R}}_{\mathsf{DTstat}}$.*

Derive(srs, (S,m)):

Compute $\mathsf{srs}_{(S,m)} \leftarrow \mathsf{CP}_{\mathsf{stat}}.\mathsf{Derive}(\mathsf{ck},(S,m))$, $\mathsf{srs}_m \leftarrow \mathsf{CP}_{\mathsf{lookup}*}.\mathsf{Derive}(\mathsf{ck},N_{\mathsf{tot}},d,m)$.
Compute $\mathsf{srs}_{(B,m,d)} \leftarrow \mathsf{CP}_{\mathsf{range}}.\mathsf{Derive}(\mathsf{ck},(B,m,d))$ with values B,d contained in srs.
Return the specialized SRSs.

Prove(srs, $(\mathsf{c}_\mathsf{T},y,(\mathbf{x}_j)_{j\in[m]}),(\mathsf{T},\rho_\mathsf{T})$):

Parse $\mathsf{c}_\mathsf{T}=(\mathsf{c}_\_,\mathsf{c}_\_,\mathsf{c}_v)$ and $\rho_\mathsf{T}=(\rho_\_,\rho_\_,\rho_v)$.
Let $k_i=k_\mathsf{T}(\mathbf{x}_i)$ and $K=\{k_1,\ldots,k_m\}$, where $k_\mathsf{T}(\cdot)$ as defined in definition 6.8.
Compute matrix commitments $\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3$ to the matrices $(\bar{\mathbf{F}}_\_)_{|K},(\bar{\mathbf{F}}_\_)_{|K},\mathbf{v}_{|K}$.
Compute a proof π_{zklookup} that

$$(m;\varepsilon;((\bar{\mathbf{F}}_\_,\bar{\mathbf{F}}_\_,\mathbf{v}),(\bar{\mathbf{F}}_\_)_{|K},(\bar{\mathbf{F}}_\_)_{|K},\mathbf{v}_{|K}))\in\hat{\mathcal{R}}_{\mathsf{lookup}*}.$$

Compute a (not hiding) commitment to the matrix $\mathbf{X}$ whose rows are the vectors $(\mathbf{x}_j)_{j\in[m]}$.
Compute proofs $\pi^{\leftarrow}_{\mathsf{range}}$ and $\pi^{\rightarrow}_{\mathsf{range}}$ for the following two statements:

$$((B,m,d);\mathbf{X}-(\bar{\mathbf{F}}_\_)_{|K})\in\hat{\mathcal{R}}_{\mathsf{range}},\qquad((B,m,d);(\bar{\mathbf{F}}_\_)_{|K}-\mathbf{X}-\mathbf{1})\in\hat{\mathcal{R}}_{\mathsf{range}}.$$

Compute a proof π_{stat} that $((S,m);y;\mathbf{v}_{|K}))\in\hat{\mathcal{R}}_{\mathsf{stat}}$.
Return $(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3,\pi_{\mathsf{zklookup}},\pi^{\leftarrow}_{\mathsf{range}},\pi^{\rightarrow}_{\mathsf{range}},\pi_{\mathsf{stat}})$.

Verify(srs, $\mathsf{vk}_{(S,m)},(\mathsf{c}_\mathsf{T},y,(\mathbf{x}_j)_{j\in[m]}),\pi_\mathsf{T}$):

Parse the proof $\pi_\mathsf{T}=(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3,\pi_{\mathsf{zklookup}},\pi^{\leftarrow}_{\mathsf{range}},\pi^{\rightarrow}_{\mathsf{range}},\pi_{\mathsf{stat}})$.
Compute $\mathsf{c}_\mathbf{X}\leftarrow\mathsf{Com}(\mathsf{ck},\mathbf{X})$ where $\mathbf{X}$ is the matrix whose rows are the vectors $(\mathbf{x}_j)_{j\in[m]}$. Return 1 if the following statements hold (else 0):

1. $\mathsf{CP}_{\mathsf{lookup}*}.\mathsf{Verify}(\mathsf{srs},\mathsf{vk}_m,(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3,\mathsf{c}_\_,\mathsf{c}_\_,\mathsf{c}_v),\pi_{\mathsf{zklookup}})=1$
2. $\mathsf{CP}_{\mathsf{range}}.\mathsf{Verify}(\mathsf{srs},\mathsf{vk}_{(B,m,d)},(\mathsf{c}_\mathbf{X}-\mathsf{c}_1),\pi^{\leftarrow}_{\mathsf{range}})=1$ and
   $\mathsf{CP}_{\mathsf{range}}.\mathsf{Verify}(\mathsf{srs},\mathsf{vk}_{(B,m,d)},(\mathsf{c}_2-\mathsf{c}_\mathbf{X}-[1]_1),\pi^{\rightarrow}_{\mathsf{range}})=1$
3. $\mathsf{CP}_{\mathsf{stat}}.\mathsf{Verify}(\mathsf{srs},\mathsf{vk}_{(S,m)},(\mathsf{c}_3,y),\pi_{\mathsf{stat}})=1$
4. $\mathsf{VerCom}(\mathsf{ck},\mathsf{c}_\mathsf{T})=1$.

Figure 6.4: Our CP-SNARK CP_{DT}. The pre-processing algorithm runs the preprocessing of the matrix lookup argument on $\bar{\mathbf{F}}_\_,\bar{\mathbf{F}}_\_,\mathbf{v}$ and openings $\rho_T=(\rho_\_,\rho_\_,\rho_v)$.

Table 6.2: Comparison between our solution and [11] for zero-knowledge decision tree accuracy. Parameters are d (number of attributes), m (size of sample), $|\mathcal{H}|$ is the cost of hash function invocation (such as SHA256); $|\mathcal{H}_{\text{circ}}|$ is the cost of a hash function invocation as a circuit; $\mathbb{P}$ is the cost of one pairing. Notation $\tilde{O}(f)$ refers to $O(f \log f)$. This table does not include the one-time cost of preprocessing for the prover (see table 6.1 for concrete costs). Notice that the asymptotics in the row for our construction account for just the commitment algorithm and the *extractability* proof. The asymptotics reported for [11] are actually a lower bound and do not include some additional factors in their complexity, such as tree height. Dominated factors, such as B and k (input and output size of decision tree respectively), are also not included in the asymptotics.

Scheme	Commit Time	Prover Time	Verifier Time	Proof Size
[11]	$O(N_{\text{tot}})\|\mathcal{H}\|$	$\tilde{O}(md + N_{\text{tot}} \log m + N_{\text{tot}}\|\mathcal{H}_{\text{circ}}\|)\mathbb{F}$	$O(md)\mathbb{F}$	$O(\log^2(md))\mathfrak{f}$
Our solution	$\tilde{O}(dN_{\text{tot}})(\mathbb{G}+\mathbb{F})$	$\tilde{O}(md)(\mathbb{G}+\mathbb{F})$	$O(m)\mathbb{G} + O(1)\mathbb{P}$	$O(1)(\mathfrak{g}_1 + \mathfrak{f})$

6.6.5. EFFICIENCY AND CONCRETE INSTANTIATIONS

We discuss how to instantiate our scheme above, the resulting system has a universal trusted setup.

- $\mathsf{CP}_{\mathsf{lookup}*}$ can be instantiated with our construction mtx[zkcq$^+$] from section 6.3;
- $\mathsf{CP}_{\mathsf{range}}$ can be implemented through a (vector) lookup in a table of size B where the subvector being looked up is of size m [9];
- $\mathsf{CP}_{\mathsf{stat}}$ can be implemented through a general-purpose commit-and-prove SNARK, such as [16, 38]. For concreteness, and to minimize proof size, in the remainder of this document, we consider the proof scheme CP-LunarLite from [16] (Section 9.4).

We can provide an upper bound on the total proof size for the instantiations above to $20\mathbb{G}_1$ elements[10] per each of the proof above (this is a loose upper bound)—see table 6.1 in this work, Table 1 and Section 9.4 in [16]. On a concrete curve like BLS12-381 this yields a total proof size of *at most* approximately 3.84KB (this is a generous lower bound). For comparison, the proof size in [11] is of the order of hundreds of kilobytes.

Decision Tree Accuracy. In the specific case of proving decision tree accuracy we prove that a decision tree is able to correctly estimate a specific fraction of a given data sample. Namely we consider the statistic that upon input $(v_j)_{j\in[m]}, (y_j)_{j\in[m]}$ computes $\sum_j \mathsf{eq}_k(v_j, y_j)/m$, $v_j = \mathsf{T}(\mathbf{x}_j)$ for $j \in [m]$ where $k \in \mathbb{N}$ is a small constant and eq_k is the function returning 1 when its two arguments, of size k, are equal[11]; otherwise it returns 0. Thanks to theorem 6.8 this can be reduced to a CP-SNARK for the following relation[12]:

$$\mathcal{R}_{\mathsf{acc}} = \left\{ (m,k); \left((y_j)_{j\in[m]}, n^*\right); \left(v_j\right)_{j\in[m]} : n^* = \sum_j \mathsf{eq}_k\left(v_j, y_j\right) \right\} \tag{6.27}$$

Even with an R1CS-based (Rank-1 Constraint System) general purpose SNARK, the relation above can be implemented very efficiently.

[9] The idea is to consider the table $\mathbf{b} = (j)_{j\in[B]}$ and prove, through a lookup argument, that that $\bar{\mathbf{x}} < \mathbf{b}$ where $\bar{\mathbf{x}}$ is the vectorization of $\mathbf{X}$.

[10] We approximate the size of field elements with that of $\mathbb{G}_1$ elements.

[11] In typical applications of decision trees the labels are integer values belonging to a small domains, for example, either booleans or bytes.

[12] Here expressed as a sum instead of a fraction. Since the size of the sample is public this is equivalent.

Our estimates show improvements of almost one order of magnitude for proving time and two orders of magnitude for verification time for representative choices of parameters (see appendix A.7 for details). Our prover runs in the order of a few seconds; our verifier in the order of 100ms. The construction in [11] in contrast has a prover running in the order of minutes (2-5m) and a verifier running in the order of 10s[13].

[13] These estimates refer to running times on an AWS EC2 c5.9xlarge. This architecture is comparable to the one used in [11].

REFERENCES

[1] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza. "SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge". In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. Vol. 8043. Lecture Notes in Computer Science. Springer, 2013, pp. 90–108.

[2] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge". In: *IACR Cryptol. ePrint Arch.* (2019), p. 953.

[3] J. Bootle, A. Cerulli, J. Groth, S. K. Jakobsen, and M. Maller. "Arya: Nearly Linear-Time Zero-Knowledge Proofs for Correct Program Execution". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*. Vol. 11272. Lecture Notes in Computer Science. Springer, 2018, pp. 595–626.

[4] L. Eagen, D. Fiore, and A. Gabizon. "cq: Cached quotients for fast lookups". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1763.

[5] J. Posen and A. A. Kattis. "Caulk+: Table-independent lookup arguments". In: *IACR Cryptol. ePrint Arch.* (2022), p. 957.

[6] A. Zapico, V. Buterin, D. Khovratovich, M. Maller, A. Nitulescu, and M. Simkin. "Caulk: Lookup Arguments in Sublinear Time". In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*. ACM, 2022, pp. 3121–3134.

[7] A. Zapico, A. Gabizon, D. Khovratovich, M. Maller, and C. Ràfols. "Baloo: Nearly Optimal Lookup Arguments". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1565.

[8] A. Gabizon and Z. J. Williamson. "plookup: A simplified polynomial protocol for lookup tables". In: *IACR Cryptol. ePrint Arch.* (2020), p. 315.

[9] A. Arun, S. T. V. Setty, and J. Thaler. "Jolt: SNARKs for Virtual Machines via Lookups". In: *IACR Cryptol. ePrint Arch.* (2023), p. 1217.

[10] A. Kate, G. M. Zaverucha, and I. Goldberg. "Constant-Size Commitments to Polynomials and Their Applications". In: *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 177–194.

[11] J. Zhang, Z. Fang, Y. Zhang, and D. Song. "Zero Knowledge Proofs for Decision Tree Predictions and Accuracy". In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. ACM, 2020, pp. 2039–2053.

[12] U. Haböck. "Multivariate lookups based on logarithmic derivatives". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1530.

[13] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. "Aurora: Transparent Succinct Arguments for R1CS". In: *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*. Vol. 11476. Lecture Notes in Computer Science. Springer, 2019, pp. 103–128.

[14] B. Chen, B. Bünz, D. Boneh, and Z. Zhang. "HyperPlonk: Plonk with Linear-Time Prover and High-Degree Custom Gates". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part II*. Vol. 14005. Lecture Notes in Computer Science. Springer, 2023, pp. 499–530.

[15] H. Chen, H. Zhang, S. Si, Y. Li, D. S. Boning, and C. Hsieh. "Robustness Verification of Tree-based Models". In: *NeurIPS 2019*. Red Hook, NY, USA: Curran Associates, Inc., Dec. 2019, pp. 12317–12328.

[16] M. Campanelli, A. Faonio, D. Fiore, A. Querol, and H. Rodríguez. "Lunar: A Toolbox for More Efficient Universal and Updatable zkSNARKs and Commit-and-Prove Extensions". In: *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III*. Vol. 13092. Lecture Notes in Computer Science. Springer, 2021, pp. 3–33.

[17] J. Groth. "On the Size of Pairing-Based Non-interactive Arguments". In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 305–326.

[18] H. Lipmaa, J. Siim, and M. Zajac. "Counting Vampires: From Univariate Sumcheck to Updatable ZK-SNARK". In: *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part II*. Vol. 13792. Lecture Notes in Computer Science. Springer, 2022, pp. 249–278.

[19] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM, 2019, pp. 2111–2128.

[20] C. Ràfols and A. Zapico. "An Algebraic Framework for Universal and Updatable SNARKs". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, pp. 774–804.

[21] S. T. V. Setty. “Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup”. In: *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*. Vol. 12172. Lecture Notes in Computer Science. Springer, 2020, pp. 704–737.

[22] S. T. V. Setty, J. Thaler, and R. S. Wahby. “Unlocking the lookup singularity with Lasso”. In: *IACR Cryptol. ePrint Arch.* (2023), p. 1216.

[23] A. R. Choudhuri, S. Garg, A. Goel, S. Sekar, and R. Sinha. “SublonK: Sublinear Prover PlonK”. In: *IACR Cryptol. ePrint Arch.* (2023), p. 902.

[24] R. E. Ali, J. So, and A. S. Avestimehr. “On Polynomial Approximations for Privacy-Preserving and Verifiable ReLU Networks”. In: *CoRR* abs/2011.05530 (2020). arXiv: 2011.05530.

[25] B. Feng, L. Qin, Z. Zhang, Y. Ding, and S. Chu. “ZEN: An Optimizing Compiler for Verifiable, Zero-Knowledge Neural Network Inferences”. In: *IACR Cryptol. ePrint Arch.* (2021), p. 87.

[26] D. Kang, T. Hashimoto, I. Stoica, and Y. Sun. “Scaling up Trustless DNN Inference with Zero-Knowledge Proofs”. In: *CoRR* abs/2210.08674 (2022). arXiv: 2210.08674.

[27] S. Lee, H. Ko, J. Kim, and H. Oh. “vCNN: Verifiable Convolutional Neural Network Based on zk-SNARKs”. In: *IEEE Transactions on Dependable and Secure Computing* (2023), pp. 1–17.

[28] T. Liu, X. Xie, and Y. Zhang. “zkCNN: Zero Knowledge Proofs for Convolutional Neural Network Predictions and Accuracy”. In: *CCS ’21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 2021, pp. 2968–2985.

[29] H. Wang and T. Hoang. “ezDPS: An Efficient and Zero-Knowledge Machine Learning Inference Pipeline”. In: *Proc. Priv. Enhancing Technol.* 2023.2 (2023), pp. 430–448.

[30] J. Weng, J. Weng, G. Tang, A. Yang, M. Li, and J. Liu. “pvCNN: Privacy-Preserving and Verifiable Convolutional Neural Network Testing”. In: *IEEE Trans. Inf. Forensics Secur.* 18 (2023), pp. 2218–2233.

[31] J. Groth, M. Kohlweiss, M. Maller, S. Meiklejohn, and I. Miers. “Updatable and Universal Common Reference Strings with Applications to zk-SNARKs”. In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*. Vol. 10993. Lecture Notes in Computer Science. Springer, 2018, pp. 698–728.

[32] M. Campanelli, D. Fiore, and A. Querol. “LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM, 2019, pp. 2075–2092.

[33] S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. “On the Non-malleability of the Fiat-Shamir Transform”. In: *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*. Vol. 7668. Lecture Notes in Computer Science. Springer, 2012, pp. 60–79.

[34] C. Ganesh, C. Orlandi, M. Pancholi, A. Takahashi, and D. Tschudi. “Fiat-Shamir Bulletproofs are Non-Malleable (in the Algebraic Group Model)”. In: *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*. Vol. 13276. Lecture Notes in Computer Science. Springer, 2022, pp. 397–426.

[35] G. Fuchsbauer, E. Kiltz, and J. Loss. “The Algebraic Group Model and its Applications”. In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*. Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 33–62.

[36] A. Chiesa, Y. Hu, M. Maller, P. Mishra, P. Vesely, and N. P. Ward. “Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS”. In: *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*. Vol. 12105. Lecture Notes in Computer Science. Springer, 2020, pp. 738–768.

[37] H. Lipmaa, R. Parisella, and J. Siim. “Algebraic Group Model with Oblivious Sampling”. In: *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part IV*. Vol. 14372. Lecture Notes in Computer Science. Springer, 2023, pp. 363–392.

[38] D. F. Aranha, E. M. Bennedsen, M. Campanelli, C. Ganesh, C. Orlandi, and A. Takahashi. “ECLIPSE: Enhanced Compiling Method for Pedersen-Committed zkSNARK Engines”. In: *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part I*. Vol. 13177. Lecture Notes in Computer Science. Springer, 2022, pp. 584–614.

7

DISCUSSIONS

Since 2020, when the *Spark! Living Lab* project started, we have investigated different use cases in collaboration with other institutions and companies. We first focused on logistics-related use cases such as bin-packing and location sharing. With the progress of the project, AI is developing rapidly and is being applied in almost all domains, which also include supply chain management. We then moved our focus to AI privacy protection, not only for supply chain management but also for AI privacy in general.

The previous chapters include our designed solutions for critical privacy issues in data anonymization for bin-packing, privacy-preserving location data sharing, and privacy-preserving machine learning in stages of data processing, data management, and data analysis. We utilize differential privacy and cryptographic protocols to achieve lower computation complexity and stronger privacy guarantees. Moreover, we take advantage of blockchain for data sharing and k-anonymity for data anonymization. This chapter first summarizes the contribution of the thesis and explains how the research questions are answered according to the use cases as discussed in Chapter 1. After that, we discuss the limitations of our proposals in three parts: the utility-privacy trade-off for differential privacy, the potential of blockchain in supply chains, and the real-world deployment of our proposals. Also, we include the potential future works.

7.1. SUMMARY OF CONTRIBUTIONS

In this section, we summarize our contributions according to the three use cases and five research questions addressed in the thesis.

7.1.1. DATA ANONYMIZATION FOR BIN-PACKING

For this use case, we investigate the following question in Chapter 2:

Q1: *How to increase efficiency in privacy-preserving bin-packing?*

As discussed in Section 1.3, the objective is to increase the efficiency (in terms of anonymization time and space usage) of the data anonymization algorithm for privacy-

preserving bin packing. We propose two different privacy-preserving data publishing approaches using differential privacy to solve bin-packing problems under privacy-preserving. By calculating the probability of identifying the correct item, we prove that both proposed methods can provide better privacy guarantees than the previous work using k-anonymity. Using differential privacy, each item is supposed to be hidden among a group of items instead of only k items by using k-anonymity. Also, we carry out seven different experiments based on different data distributions and a different number of inputs. The results show that our proposed methods are much faster than the k-anonymous approach (from 10^3 s to less than 0.1 s) without any cost of objective ratio (to evaluate the number of trucks) or feasibility (to evaluate the probability of overloading). Both proposed methods show advantages in privacy preservation and run-time over existing approaches that only apply k-anonymity or differential privacy while showing comparable objective ratios and feasibility. Meanwhile, both proposed methods can be extended to solve 2-D or 4-D bin-packing problems

When we apply privacy-preserving methods, a better privacy guarantee always means a less useful output, so it is important to find the trade-off between these two aspects. In this paper, we use experiments to show the relationship between privacy guarantees (k and ϵ) and performance (o/o_n and f). With some performance cost (10%−20% o/o_n and f), the proposed methods can provide good privacy guarantees (such as $\epsilon = 1$). A better utility function or a better clustering method can help improve the performance of both proposed methods. We give further discussions about the privacy-utility trade-off in Section 7.2.1.

7.1.2. PRIVACY-PRESERVING LOCATION DATA SHARING

As discussed in Section 1.3, privacy-preserving location data sharing involves two parts: location data perturbation and privacy-preserving data sharing.

For location data perturbation, we address the following question in Chapter 3:

Q2: *How to provide location privacy for tracking services for logistics in practice?*

The objective is to de-identify trajectory data in practice while keeping utility for privacy-preserving location-based services. We use differential privacy and geo-indistinguishability with different privacy levels for corresponding receivers. Our concrete privacy analysis and proof indicate the proposed angle selection algorithm can provide better trajectory privacy preservation under real road maps and possible attacks than existing works. The detailed experiments show how privacy parameters are selected and how the utility remains in terms of arrival prediction. Also, the run-time is in the order of nanoseconds, which is feasible for real-time data sharing.

For privacy-preserving location data sharing, we address the following question:

Q3: *How to design a privacy-preserving and decentralized platform for location sharing with constrained IoT devices?*

In Chapter 3, we design a blockchain-based framework to share location data in a privacy-preserving manner among multiple logistic companies and users. Our proposed location sharing protocols can protect privacy-sensitive data using cryptographic con-

structions under centralized and decentralized settings. Our security analysis proves that the system is privacy-preserving. With Ethereum, our proposal has lower storage costs compared to the previous work [1]. It is feasible and can handle ~450 trucks at the same time, which is a reasonable amount for an average city. Also, companies can build their own solutions using our protocols to improve.

For further improvement, in Chapter 4, we present **PrivTrack**, a general framework for privacy-preserving location data sharing. We leverage our differentially private trajectory perturbation algorithm and cryptographic protocols for privacy-preserving trajectory tracking for blockchain-based supply chains with constrained IoT devices. Now, location privacy with constrained IoT devices is available to track the trajectory of trucks for delivery in big cities. With constrained IoT devices, we first show that our trajectory perturbation algorithm provides privacy protection against median filter attacks under real road maps by misleading the adversary to a wrong trajectory. The output perturbed trajectory data is useful for eCMR and package tracking. Also, our evaluation illustrates the feasibility of applying our trajectory perturbation algorithm in constrained IoT devices. Then, we propose a platform that enables the interaction of constrained devices with a blockchain-based platform, offering data validation, authenticity and access control. Our security and privacy analysis show that the proposed protocols provide privacy-preserving data sharing together with the trajectory perturbation algorithm. Based on our experiments, we demonstrate that our protocols have a low impact on the IoT device battery life. When using an IoT device based on an Arm Cortex-M0 processor, the run time is also feasible: **3.6 ms** when the random numbers are generated in software, and **76.6 ms** when generated in a NIST-recommended hardware module. Considering the overall performance of constrained IoT devices, the run time of our protocols is ~200 ms with an autonomy of almost one month, which is well aligned with the needs of real-world use cases.

7.1.3. PRIVACY-PRESERVING MACHINE LEARNING

As discussed in Section 1.3, for privacy-preserving machine learning, we focus on privacy-preserving small-scale collaborative learning and decision tree model validation.

For privacy-preserving small-scale collaborative learning, we investigate the following question in Chapter 5:

Q4: *How to design a privacy-preserving small-scale collaborative learning system against inference attacks?*

In our use case, the performance of automated fraud detection is hindered by a lack of positive samples in collected data. While this problem could be solved by using sequential collaborative learning, its vulnerability to powerful inference attacks restricts its applicability. In Chapter 5, we present a protocol to prevent the execution of inference attacks using secure multi-party computation techniques. To achieve this, we require parties to jointly determine the training order. While doing so, we ensure that participants only receive information on whom to send their data to. This disables them from training a model using parameters received from a specific target participant, thereby preventing the execution of inference attacks. By including a security analysis, we have shown our protocol is robust against semi-honest participants and leaders. Addition-

ally, we discuss how to make the protocol resistant to timing attacks using differential privacy. With this work, we contribute a practical protocol that is robust against inference and timing attacks to facilitate privacy-preserving sequential collaborative learning. To our knowledge, our work is the first to prevent inference attacks using a secure joint permutation selection protocol with an overhead of only a few seconds.

For decision tree model validation, we investigate the following question:

Q5: *How to validate the correctness and performance of a private decision tree in a privacy-preserving manner?*

In Chapter 6, we give a novel application of zero-knowledge matrix lookup argument to the domain of zero-knowledge decision tree where the model provider releases a commitment to a decision tree and can prove zero-knowledge statistics over the committed data structure. We improve over the framework of Zhang *et al.* [2], which shows zkSNARKs for evaluations of committed decision trees and zkSNARKs for accuracy of committed decision trees. The former kind of zero-knowledge protocols can prove that a committed decision tree T, on input from a vector $\mathbf{x}$, outputs a label v, while the latter schemes enable the validation of the accuracy (namely, the ratio of true positives) of a decision tree on a given dataset. Our framework can instantiate different kinds of statistics over the committed decision trees, including evaluation and accuracy. Our design decouples the computation of the committed decision tree and the performed statistics. This allows for a plug-and-play approach. For security, we extend the notion of security from [2] considering possibly maliciously generated commitments to decision trees. Our scheme based on lookup arguments has succinct verification, the prover's time complexity is asymptotically better than the state of the art, and it is secure in a strong security model where the commitment to the decision tree can be malicious.

7.2. LIMITATIONS AND FUTURE WORKS

In this section, we further discuss the limitations and future works of the thesis. Based on our contributions and works in previous chapters, this section includes three main topics for discussion: the utility-privacy trade-off for differential privacy in practice, the potential of blockchain in supply chains, and the real-world deployment of our proposed solutions.

7.2.1. DIFFERENTIAL PRIVACY IN PRACTICE

In Chapters 2, 3 and 5, differential privacy is utilized as a main tool for data privacy enhancement, but there is a lack of a good way of quantifying 'privacy' for differential privacy in practice.

In Chapter 2, though ϵ and k provide parameters to quantify privacy protection levels for data anonymization, it is still unclear which level is a 'good' level for privacy protection against adversaries. For instance, "$\epsilon = 1$" provides different practical privacy protection levels with different schemes and scenarios. In data anonymization, the privacy protection level is not only determined by ϵ but also by the statistics of the dataset, making it harder to add an appropriate amount of noise to achieve privacy protection. One possible solution is to deploy linkage attacks with different ϵ values to find a good utility-

privacy trade-off. However, this is time-consuming in practice and may not counter other unknown attacks. Based on the findings, there remains more work on a fairer way of quantifying privacy protection other than only using the values of ϵ from differential privacy and k from k-anonymity. A fair privacy protection metric, together with the existing information loss metrics, is beneficial for finding a better trade-off between privacy and utility.

Similar to Chapter 2, in Chapter 5, it is difficult to choose a suitable ϵ under differential privacy to achieve a 'good' level of privacy during the training to protect against the inference attacks. Many researchers usually pay more attention to finding a new notion of differential privacy or achieving a tighter boundary by adding noise differently. However, when companies claim that they publish a model with $\epsilon = 1$, it is still unclear whether the model is published in a privacy-preserving manner against possible attacks since privacy is also determined by the statistics of datasets and how the model is trained. This brings the difference between theory and practice. In theory, applying differential privacy can provide privacy guarantees against inference attacks. However, in practice, there are existing works with ϵ values as large as 100 or 1000, which makes the application of differential privacy not meaningful any more. It is also vital that techniques are not misused since this makes people believe that an insecure system is secure. This is even more horrible than no privacy-preserving methods are applied. Similar to data anonymization, ideally, there could be a better way to represent privacy protection level instead of only using ϵ. In the meantime, in Chapter 5, though differential privacy provides provable privacy guarantees with low computation cost, it negatively influences model utility with the introduced noise. Instead, secure multi-party computation provides strong privacy guarantees while maintaining data utility with higher computation complexity. In Chapter 5, we combine the use of differential privacy and cryptographic protocols to protect privacy among small-scale participants against inference attacks. The proposal implies that neither differential privacy nor cryptographic tools are perfect for optimizing all factors. Further studies remain on how to utilize differential privacy and cryptographic protocols in a more efficient manner to provide privacy protection for various scenarios and use cases.

In Chapter 3, we propose a scheme that provides weaker theoretical privacy guarantees but better practical privacy protection. However, the focus here is not all about whether a scheme should be evaluated in a more practical setting. The core of our idea is to be more practice-oriented instead of only considering privacy guarantees in theory. Taking trajectory hiding, for instance, our focus is on how to de-identify on which road the vehicles or pedestrians are moving. It is important to achieve tighter boundaries or define new notions for location privacy based on differential privacy, but more attention is needed to whether this contribution benefits practical uses. In Chapter 3, we assume the adversary holds the city road map and applies filter attacks, which is the first work to address trajectory hiding in a practical scenario by misleading adversaries to identify a wrong road or trajectory. However, there remains a probability of a correct guess (due to the randomness of differential privacy), and it is still unknown whether there are other trajectory reconstruction attacks. Possible approaches, such as road selection and location perturbation, can further enhance the practical privacy protection of existing solutions based on road maps. Also, the investigation of other trajectory reconstruction

methods, such as machine learning or graph-based methods, remains in future studies.

7.2.2. BLOCKCHAIN IN SUPPLY CHAINS

In Chapters 3 and 4, blockchain is utilized as a good choice of platform for privacy-preserving data sharing. At the beginning stage of the *Spark! Living Lab* project, blockchain is seen as a disruptive technology for supply chains. However, it turns out that blockchain has few advantages for supply chain use cases.

In both Chapter 3 and Chapter 4, we introduce blockchain-based privacy-preserving data sharing platforms for location data. Based on the literature, blockchain is a potentially disruptive technology for supply chains due to the properties of decentralization, traceability, immutability and transparency [3]. In supply chains, such properties are desired. Especially for small and medium-sized enterprises (SMEs), a decentralized platform makes it possible to have a shared supply chain platform with others to lower the cost of digitalization. Meanwhile, traceability enables companies to track and trace their raw materials. Immutability and transparency also build trust among companies and customers. In theory, it seems a perfect match between blockchain and supply chain. Unfortunately, there are only limited existing applications of blockchain for a decentralized solution for supply chains, and blockchain faces different kinds of challenges when deployed in practice, such as data validation, privacy, and adoption issues.

In Chapter 4, we propose a solution for data validation by removing human interaction in the process. Our proposed approach works well according to our analysis and experiments, but it does not mean that companies are willing to use the proposal. Companies are reluctant to share data with others because commercially sensitive information may be leaked. Instead, companies prefer a more centralized system where they can control all the data internally. Furthermore, companies usually lack expertise in blockchain, making it difficult to build their own blockchain-based solution. This means companies need to pay even more to replace the current supply chain system with a blockchain-based solution. As a result, companies are interested in blockchain since they do not want to miss out on any opportunity for the application of emerging technology rather than having a use case where blockchain is a perfect match. In fact, most companies do not have clear ideas about how blockchain should be applied.

Meanwhile, blockchain is not the only solution for supply chains. Other distributed solutions, such as distributed databases, also offer potential solutions for data sharing and storage [4, 5]. Another weakness of blockchain is the computational complexity for consensus and energy consumption, which also raises costs and lowers scalability. New solutions based on proof-of-stake provide new opportunities, but they do not have obvious advantages over other techniques. Companies can use blockchain as one of the possible solutions, but it is not necessarily the best. In supply chains, there exist only a few suitable use cases for blockchain, and blockchain only has advantages over other approaches in limited scenarios. Except for supply chains, the adoption of blockchain in other domains, such as healthcare, economics and education, remains further multidisciplinary research to investigate whether blockchain is a suitable solution or a hype.

7.2.3. DEPLOYMENT OF OUR PROPOSALS

The real-world deployment of our proposals can have a significant impact, but unfortunately, this does not happen. As discussed in Section 7.2.2, companies do not want to deploy blockchain-based solutions since they want to keep control of their supply chains, but there are more reasons that hinder the deployment of our proposals.

One reason is the difference between the assumed adversary model and the adversary in practice. For example, in Chapter 6, though our proposal can avoid the verifier accessing the decision tree structure, it is still possible for the adversary to rebuild the model based on several rounds of interaction. The adversary can feed the model owner fine-tuned data samples and infer the decision boundaries based on the outputs from the decision tree. With the stolen model, the adversary can further infer the training set of the model. There is a lack of trust among different parties, and there needs to be more work to improve trust further. One potential way is to combine different security and privacy approaches together to advance privacy protection, such as zero-knowledge proof and differential privacy. Another way is to build more explainable AI models and support the model performance with a solid theoretical background.

Besides the reasons above, another main factor is the environment. In the *Spark! Living Lab* project consortium, there are different logistic companies that brought multiple use cases. Our work brings privacy protection solutions closer to practice using differential privacy and cryptographic tools. It was a good opportunity to deploy our design and proposals in real. However, due to COVID-19 starting in 2020, global supply chains are affected in terms of human resources and transportation. For companies, there is no urgent need to deploy privacy protection proposals since they do not lead to competitive advantages. Though our proposals are well recognized by our company partners in the project, they are not a game changer for the business. Instead, the investment-profit trade-off of our proposals is not convincing for the companies. Enhancing privacy protection can strengthen the trust between companies and customers, which has long-term benefits in terms of profits and reputation. However, privacy enhancement requires substantial initial investment costs when deploying advanced techniques. According to the study [6], the budget for GDPR compliance can be $50 million for a single company. Moreover, companies are hesitant to incur investment costs since the reward is often implicit and realized over extended periods. Consumers are also willing to trade their privacy for more convenient services [7]. Furthermore, the hype of blockchain in supply chains also makes companies more careful when considering the application of research outputs in practice. To advance the deployment of privacy protection proposals in companies, one way is to enforce privacy laws and regulations, which ensures that companies have to prioritize privacy enhancement. Another way is to improve social consciousness of the importance of privacy. When customers have more privacy concerns, companies need to enhance privacy protection to maintain good reputations and retain their customers.

Nevertheless, there are potentials and opportunities. Recently, with the rapid development and application of AI in different domains, companies are putting more emphasis on the application of machine learning models to benefit their business. Subsequently, AI privacy has become more important than ever since a machine learning model can potentially expose sensitive information about the training data. Meanwhile,

a private model can have a high value, so it is also essential to deal with the ownership and verification issues. Now, machine learning is usually being used without much privacy protection or privacy guarantees, as people are more focused on the utility and performance of a model. Further development of AI requires further investigation and deployment of privacy techniques to avoid possible leakage and commercial loss.

In this thesis, we focus on enhancing privacy preservation in the training and deployment of machine learning models. We propose a framework to mitigate inference attacks in collaborative learning settings. Additionally, we introduce the zero-knowledge lookup arguments for private decision tree evaluation. Both contributions advance data privacy protection in machine learning and promote the development of more secure and privacy-preserving machine learning systems.

REFERENCES

[1] M. E. Maouchi, O. Ersoy, and Z. Erkin. “DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain”. In: *SAC 2019*. ACM, 2019, pp. 364–373.

[2] J. Zhang, Z. Fang, Y. Zhang, and D. Song. “Zero Knowledge Proofs for Decision Tree Predictions and Accuracy”. In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. ACM, 2020, pp. 2039–2053.

[3] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen. “Blockchain technology and its relationships to sustainable supply chain management”. In: *Int. J. Prod. Res.* 57.7 (2019), pp. 2117–2135.

[4] T. C. Du, H. Lee, and A. Chen. “Constructing federated databases in coordinated supply chains”. In: *Decis. Support Syst.* 36.1 (2003), pp. 49–64.

[5] D. E. O'Leary. “Some issues in blockchain for accounting and the supply chain, with an application of distributed databases to virtual organizations”. In: *Intell. Syst. Account. Finance Manag.* 26.3 (2019), pp. 137–149.

[6] S. Sirur, J. R. C. Nurse, and H. Webb. “Are We There Yet?: Understanding the Challenges Faced in Complying with the General Data Protection Regulation (GDPR)”. In: *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security, MPS@CCS 2018, Toronto, ON, Canada, October 15, 2018*. ACM, 2018, pp. 88–95.

[7] P. A. Norberg, D. R. Horne, and D. A. Horne. “The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors”. In: *Journal of Consumer Affairs* 41.1 (2007), pp. 100–126.

A

SUPPLEMENTARY MATERIAL FOR CHAPTER 6

A.1. ON APPLYING OTHER BACKENDS

Here we elaborate on why applying different backends (e.g., Groth16 [1] or Marlin [2]) does not substantially change our comparison with [3]. This is particularly true for proving time, for which the main bottleneck of the approach in [3] lies in the number of constraints, which stems from the essence of their "tree-visiting" approach. This results in dependence on the size of the hash function among other metrics, as discussed in the introduction.

We verify this claim by running [3] on Groth16—one of the most optimized, highly succinct proof systems—instead of Aurora (the main backend originally described [3]) and using their original code[1]. When running it over Groth16 we observe that the scheme in [3] runs in ≈ 3 minutes. If one wanted to use a universal setup scheme (e.g., Marlin) these numbers would be at least 4x as large. As a comparison, we recall that the ballpark of our scheme is 15-30s (pg30). This results in approximately an order of magnitude difference between our scheme and Marlin applied to [3]. Note that this comparison is apple-to-apple: it is on the same architecture, on the parameters referred at 164, and using the original code for [3].

The above observations are for proving time. For other dimensions—verification and proof size—applying a scheme like Marlin to [3] would obtain numbers close to ours. For proof size these numbers may be marginally better than ours.

A.2. ADDITIONAL MATERIAL ON SECTION 6.2

Definition A.1 (Universal CP-SNARK)**.** *A universal CP-SNARK for an indexed relation $\mathcal{R}$ is a tuple of algorithms* $\Pi = (\mathsf{KGen}, \mathsf{Derive}, \mathsf{Prove}, \mathsf{Verify})$ *where*

[1] Available at https://github.com/TAMUCrypto/ZKDT_release.

- Derive *is a deterministic algorithm that takes as input an* srs *(which includes relation parameters* pp*) produced by* KGen*, an index* ind*, and outputs specialized SRS* $\mathsf{srs}_{ind} = (\mathsf{ek}_{ind}, \mathsf{vk}_{ind})$*. The length of* vk_{ind} *is* $\mathsf{poly}(n, \log|ind|)$.
- *Consider the relation* $\mathcal{R}'$ *such that* $\mathcal{R}'(\mathsf{pp}, (ind, x), w) \iff \mathcal{R}(\mathsf{pp}, ind, x, w)$*, the tuple of algorithms* $(\mathsf{KGen}, \mathsf{Prove}, \mathsf{Verify}')$ *is an argument system for the relation* $\mathcal{R}'$ *and* Verify' *is the algorithm that upon input* srs*, instance* (ind, x) *and a proof* π*, first runs* Derive *on* srs *and index* ind*, then runs* $\mathsf{Verify}(\mathsf{vk}_{ind}, x, \pi)$.

A.3. ADDITIONAL MATERIAL ON SECTION 6.4

Lemma A.1. $\sum_{j=0}^{N} A(\omega_N^{j-1}) = \sum_{i=0}^{n} B(\omega_n^{i-1})$ *iff* $\sum_{j=0}^{N} C(\omega_N^{j-1}) = 0$.

Proof. Recall that we denote $C(X) := A(X) - \frac{1}{\vartheta} B(X)\mathsf{z}(X)$, $\vartheta := N/n$ and $\mathsf{z}(X) := \nu_{\mathbb{K}\setminus\mathbb{H}}(X)$. Define

$$\Delta(X) := A(X) - C(X) = \tfrac{1}{\vartheta} B(X)\mathsf{z}(X) \ .$$

For any $j \in [N]$, $\Delta(\omega_N^{j-1}) = \frac{1}{\vartheta} B(\omega_N^{j-1})\mathsf{z}(\omega_N^{j-1})$. Now, $\mathsf{z}(\omega_N^{j-1}) = 0$ when $\omega_N^{j-1} \notin \mathbb{H}$ (i.e., $\vartheta \nmid j-1$) and $\mathsf{z}(\omega_N^{j-1}) = \vartheta$ otherwise. Hence,

$$\Delta(\omega_N^{j-1}) = \begin{cases} 0 \ , & \omega_N^{j-1} \notin \mathbb{H} \quad (\text{i.e., } \vartheta \nmid j-1) \ , \\ B(\omega_N^{j-1}) \ , & \omega_N^{j-1} \in \mathbb{H} \quad (\text{i.e., } \vartheta \mid j-1) \ . \end{cases}$$

Writing $j - 1 = (i-1)\vartheta$ in the last case, we get that when $\vartheta \mid j-1$,

$$\Delta(\omega_N^{j-1}) = B(\omega_N^{j-1}) = B(\omega_N^{(i-1)\vartheta}) = B(\omega_n^{i-1}) = B_i \ .$$

Thus,

$$\Delta(\omega_N^{j-1}) = \begin{cases} 0 \ , & \vartheta \nmid (j-1) \ , \\ B(\omega_n^{i-1}) \ , & j-1 = (i-1)\vartheta \ . \end{cases}$$

Hence, the "low-degree part" of $\Delta(X)$ only depends on the values of $B(X)$ on the subgroup $\mathbb{H}$ and hence, we do not have to perform a low-degree test on $B(X)$. Moreover, $\sum_{j=0}^{N} \Delta(\omega_N^{j-1}) = \sum_{i=0}^{n} B(\omega_n^{i-1})$. Thus, $\sum_{j=0}^{N} C(\omega_N^{j-1}) = \sum_{j=0}^{N} A(\omega_N^{j-1}) - \sum_{i=0}^{n} B(\omega_n^{i-1})$. This proves the claim. □

A.3.1. SECURITY PROOFS OF cq^+

KNOWLEDGE-SOUNDNESS.

We prove knowledge-soundness in the AGM under the standard PDL assumption. We give a complete proof for cq^+; the proof for cq^{++} follows from that and the known polynomial commitment batching lemmas.

Definition A.2 (Power Discrete Logarithm [4])**.** *Let* $d_1(n), d_2(n) \in \mathsf{poly}(n)$*. A bilinear group generator* GroupGen *is* (d_1, d_2)-PDL (Power Discrete Logarithm) secure *if for any non-uniform PPT* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{pdl}}_{d_1,d_2,\mathsf{GroupGen},\mathcal{A}}(n) :=$

$$\Pr\left[\ \mathsf{pp} \leftarrow \mathsf{GroupGen}(1^n); s \leftarrow\!\$\, \mathbb{F}^* : \quad \mathcal{A}\left(\mathsf{pp}, \left[(s^i)_{i=0}^{d_1}\right]_1, \left[(s^i)_{i=0}^{d_2}\right]_2\right) = s \ \right] = \mathsf{negl}(n) \ .$$

Theorem A.2. *Assume $U(X) \nmid X$. Let $N_1 \geq N + b - 1$ and $N_2 \geq N + 1$. Then the interactive protocol cq^+ from fig. 6.1 is knowledge-sound in the AGM under the (N_1, N_2)-PDL assumption.*

Proof. Let $\mathcal{A}$ be an algebraic knowledge-soundness adversary that, after interacting with the honest verifier, outputs the following group elements:

$$\left[m(s), A(s), B(s), Q_B(s), S(s), P(s), R_C^*(s), Q(s)\right]_1 \ ,$$

with explanations, showing that the outputs are evaluations of polynomials $m, A, B, Q_B, S, P, R_C^*, Q$ that all have degree $\leq N_1$. Moreover, $m(X)$ and $S(X)$ do not depend on β, γ, η, while $(A(X), B(X), Q_B(X))$ depend on β, and $(P(X), R_C^*(X), Q(X))$ depend on β, γ, η. The adversary also returns one field element B_γ (the claimed value of $B(\gamma)$).

As usual in the AGM proofs, one has two cases: an information-theoretic and a computational case (a reduction to PDL). One writes down a polynomial form of the verifier's equations (two equations V_1 and V_2 in the current case), such that the verifier accepts iff $V_1(s) = V_2(s) = 0$. Here, the coefficients of V_1 and V_2 can be computed from the outputs of the extractor. In the information-theoretic case, we consider the possibility that both V_1 and V_2 are zero polynomials and show that $\mathcal{A}$ must have been honest. In the computational case, we analyze the case that either $V_1(X)$ or V_2 is a non-zero polynomial, but the verifier still accepts, i.e., $V_1(s) = V_2(s) = 0$. We then construct a reduction to the security of PDL.

Information-theoretic case. By the second verification equation in fig. 6.1,

$$V_2(X) := B(X) + \eta D(X) - B_\gamma - P(X) \cdot (X - \gamma)$$

is a zero polynomial. Thus,

$$(X - \gamma) \mid (B(X) + \eta D(X) - B_\gamma) \ .$$

Since neither $B(X)$ or $D(X)$ depends on η, we get by the Schwartz-Zippel lemma that $(X - \gamma) \mid (B(X) - B_\gamma)$ and $(X - \gamma) \mid D(X)$. Hence, $B(\gamma) = B_\gamma$ and $D(\gamma) = 0$. Recalling that $D(X) = B(\gamma)(\mathsf{F}(X) + \beta) - 1 - Q_B(X)\nu_{\mathbb{H}}(\gamma)$, we get

$$B(\gamma)(\mathsf{F}(\gamma) + \beta) - 1 = Q_B(\gamma)\nu_{\mathbb{H}}(\gamma) \ .$$

Since neither $B(X)$ or $Q_B(X)$ depends on γ, by the Schwartz-Zippel lemma,

$$B(X)(\mathsf{F}(X) + \beta) - 1 = Q_B(X)\nu_{\mathbb{H}}(X)$$

as a polynomial. Matching the left and right-hand sides for the values of $X = \omega_n^{i-1}$, we get

$$B_i = \frac{1}{\mathbf{f}_i + \beta} \ .$$

Next, from the first verification equation, we get that the following polynomial $V_1(X)$ is a zero polynomial:

$$V_1(X) := \left(\begin{array}{c} A(X)\mathsf{T}(X) + (\beta + \eta) A(X) - m(X) - \\ \frac{\eta}{\vartheta} B(X)\mathsf{z}(X) + \eta^2 S(X) - Q(X)\nu_{\mathbb{K}}(X) \end{array} \right) \cdot U(X) - \eta R_C^*(X) X \ .$$

Since $U(X) \nmid X$, we get that $U(X) \mid R_C^*(X)$. Since $\deg U(X) = \mu = N_1 - N + 2$, $R_C^*(X) = R_C(X)U(X)$ for some polynomial $R_C(X)$ of degree $\leq N-2$. Thus,

$$A(X)\mathsf{T}(X) + (\beta+\eta)A(X) - m(X) - \frac{\eta}{\vartheta}B(X)\mathsf{z}(X) + \eta^2 S(X) = \eta R_C(X)X + Q(X)v_{\mathbb{K}}(X) \ .$$

Recalling that $C(X) = A(X) - \frac{1}{\vartheta}B(X)\mathsf{z}(X)$, we get

$$A(X)(\mathsf{T}(X)+\beta) - m(X) + \eta C(X) + \eta^2 S(X) = \eta R_C(X)X + Q(X)v_{\mathbb{K}}(X) \ .$$

Here, only $Q(X) = Q(X,\eta)$ and $R_C(X) = R_C(X,\eta)$ depend on η (and γ) while other polynomials do not. Let Y be the indeterminate corresponding to η, and let us write Q and R_C as bivariate polynomials in X and Y. By applying the Schwartz-Zippel lemma again,

$$A(X)(\mathsf{T}(X)+\beta) - m(X) + YC(X) + Y^2 S(X) = YR_C(X,Y)X + Q(X,Y)v_{\mathbb{K}}(X)$$

as a polynomial.

Setting $Y = 0$, we get

$$A(X)(\mathsf{T}(X)+\beta) - m(X) = Q(X,0)v_{\mathbb{K}}(X) \ .$$

Considering the values of $X = \omega_N^{j-1}$ again, we get

$$A_j = \frac{m_j}{\mathbf{t}_j + \beta} \ .$$

On the other hand, define $\widehat{Q}(X) := (Q(X,Y) - Q(X,0))/Y$ and $\widehat{r}_C(X) := (R_C(X,Y) - R_C(X,0))/Y$. Thus,

$$\begin{aligned} &A(X)(\mathsf{T}(X)+\beta) - m(X) + YC(X) + Y^2 S(X) \\ &\quad = Y\left(\widehat{r}_C(X,Y)Y + R_C(X,0)\right)\cdot X + \left(\widehat{Q}(X,Y)Y + Q(X,0)\right)\cdot v_{\mathbb{K}}(X) \ . \end{aligned}$$

Ignoring addends that do not depend on Y, we get

$$C(X) + YS(X) = \left(\widehat{r}_C(X,Y)Y + R_C(X,0)\right)\cdot X + \widehat{Q}(X,Y)v_{\mathbb{K}}(X) \ enspace.$$

Replace now Y by any constant, say $Y = 0$. Thus,

$$C(X) = R_C(X,0)X + \widehat{Q}(X,0)v_{\mathbb{K}}(X) \ .$$

Since $\deg_X R_C(X,Y) \leq N-2$, by the Aurora's sumcheck,

$$\sum_{j=1}^{N} C_j = 0 \ .$$

By theorem 6.2,

$$\sum_{j=1}^{N} A_j = \sum_{i=1}^{n} B_i \ .$$

Next, we already expressed A_j and B_i as $m_j/(\mathbf{t}_j+\beta)$ and $1/(\mathbf{f}_j+\beta)$. Thus,

$$\sum_{j=1}^{N}\frac{m_j}{\mathbf{t}_j+\beta}=\sum_{i=1}^{n}\frac{1}{\mathbf{f}_j+\beta}\ .$$

Since m_j does not depend on β, we can apply the Schwartz-Zippel lemma, obtaining

$$\sum_{j=1}^{N}\frac{m_j}{\mathbf{t}_j+X}=\sum_{i=1}^{n}\frac{1}{\mathbf{f}_i+X}\ .$$

By Haböck's lemma (theorem 6.1), this means that $\{\mathbf{f}_i\}\subseteq\{\mathbf{t}_j\}$. Hence, the adversary is honest. This proves the information-theoretical case.

Computational case. In this case, one of V_1 and V_2 is a non-zero polynomial but $V_1(s)=V_2(s)=0$. W.l.o.g., assume $V_1\neq 0$ (since $\deg V_1>\deg V_2$, this results in a stronger PDL assumption). Note that $\deg V_1\leq N_1+N+(N_1-N+2)=2(N_1+1)$. We construct the following (N_1,N_2)-PDL adversary $\mathcal{B}$. $\mathcal{B}\left(\left[(s^i)_{i=0}^{N_1}\right]_1,\left[(s^i)_{i=0}^{N_2}\right]_2\right)$ uses its as the SRS for $\mathcal{A}$. $\mathcal{B}$ then invokes $\mathcal{A}$, playing the honest verifier and obtaining a protocol transcript. Since $\mathcal{A}$ is algebraic, it returns explanations, i.e., polynomials $m,\ldots,Q$. Given these polynomials, $\mathcal{B}$ can compute all coefficients of V_1. Hence, $\mathcal{B}$ has a known non-zero polynomial V_1 of degree $2(N_1+1)$, such that $V_1(s)=0$. $\mathcal{B}$ now uses a standard probabilistic polynomial-time root-finding algorithm over finite fields to obtain all roots of V_1, and tests which root equals s by using its input. Hence, $\mathcal{B}$ can compute s and thus break (N_1,N_2)-PDL.

Analyzing both cases finishes the proof. □

Zero-Knowledge. We prove the zero-knowledge property of cq^+; the proof for cq^{++} is nearly identical and is omitted.

Theorem A.3. *The interactive protocol* cq^+ *from fig. 6.1 is honest-verifier zero-knowledge.*

Proof. To prove the theorem we first describe the simulator and then argue why its simulation is indistinguishable from a honestly generated proof.

We present the simulator in fig. A.1. What Sim does is to generate the commitments $[m(s),A(s),B(s)]_1$ as in the prover algorithm but by setting $A_j=B_i=m_j=0$. Next, it samples the remaining elements following the correct distribution that makes the verification equations accept.

Let us argue about each element, output by the simulator:

- $m(s)$ (resp. $A(s)$) is masked by random value ρ_m (resp. ρ_A); this perfectly hides $m(X)$ (resp. $A(X)$) since only one evaluation of it (at $X=s$) is known.
- $B(s)$ is masked by a degree-1 random polynomial; this perfectly hides $B(X)$ since only two evaluations of it (at $X=s$ and $X=\gamma$) are known.
- $Q_B(s)$ is computed as in the interactive protocol.
- $S(s)$ is computed as in the interactive protocol.
- $P(s)$ is computed as in the interactive protocol.

- $R_C^*(s)$ is computed as in the interactive protocol in the case $A_j = B_i = 0$. Note that the pair $(S(s), R_C^*(s))$ is uniformly random due to the choice of R_S and ρ_S.
- $Q(s)$ is chosen so that it makes the verifier accept. To check it, let us rewrite the first verification equation in cq^{++} but in discrete logarithms:

$$A(s)\mathsf{T}(s)U(s) + \left((\beta+\eta)A(s) - m(s) + \eta^2 S(s)\right)U(s) - \frac{\eta}{\vartheta}B(s)\mathsf{z}(s)U(s) - Q(s)\nu_{\mathbb{K}}(s)U(s) = \eta R_C^*(s)s$$

Writing in simulator chosen $A(X)$, $B(X)$, $m(X)$, $S(X)$, and $R_C^*(X)$, this is equivalent to

$$\rho_A(s)\nu_{\mathbb{K}}(s)\mathsf{T}(s)U(s) + \left((\beta+\eta)\rho_A(s)\nu_{\mathbb{K}}(s) - \rho_m(s)\nu_{\mathbb{K}}(s) + \eta^2\left(sR_S(s) + \nu_{\mathbb{K}}(s)\rho_S(s)\right)\right)U(s) - \frac{\eta}{\vartheta}\rho_B(s)\nu_{\mathbb{K}}(s)U(s) - Q(s)\nu_{\mathbb{K}}(s)U(s) = \eta^2 R_S(s)U(s)s$$

Cancelling $\eta^2 R_S(s)U(s)s$ and dividing the rest by $\nu_{\mathbb{K}}(s)U(X)$, it is equivalent to

$$Q(s) = (\beta+\eta+\mathsf{T}(s))\rho_A(s) - \frac{\eta}{\vartheta}\rho_B(s) - \rho_m(s) + \eta^2\rho_S(s) \ .$$

□

A.3.2. OUR FULLY ZERO-KNOWLEDGE LOOKUP ARGUMENT

We describe the protocol zkcq^+ in fig. A.2. We remark that Preproc computes the value $\tilde{\mathsf{c}}_{\mathbf{t}}$ which is included in the proof. This is just syntactic sugar, and it is only necessary for matching the syntax of CP-SNARK. In practical implementations, the value $\tilde{\mathsf{c}}_{\mathbf{t}}$ could be posted together with $\mathsf{c}_{\mathbf{t}}$, and their well-formedness could verified only once.

A.4. ADDITIONAL MATERIAL ON SECTION 6.5

Theorem 6.4. *The lookup argument* $\mathsf{mtx}[\mathsf{CP}]$ *defined in fig. 6.2 is knowledge-sound in the AGM and ROM under the* $(N\cdot d, N\cdot d)$*-PDL assumption and assuming that* CP *is knowledge-sound. Furthermore, the protocol is zero-knowledge assuming* CP *is zero-knowledge.*

Proof. Notice that the vectorization operator is linear, moreover that Preproc applies a linear function to the precomputation through CP of the vectorizations of the matrices $\mathbf{T}, \mathbf{R}, \mathbf{C}^{(N)}$. Thus the Preproc is $\mathbb{F}$-linear when Preproc′ is $\mathbb{F}$-linear.

The algorithm Prove makes $O(n\log n)$ field operations and group multiplications and additions to compute $\sigma, w, \pi_R, \pi_{R'}$ and makes $O(n)$ field operations to compute z. The proofs π_R and $\pi_{R'}$ take $O(n\log n)$ field operations and group multiplications and additions. Notice that Prove does not need to compute $\bar{\mathbf{t}}^*, \bar{\mathbf{f}}^*$ to compute the proof π^*. In fact, to compute such a proof, the prover needs only:

$$(\mathsf{aux}_j^*)_{j\in K} = (\mathsf{aux}_{M,j})_{j\in K} + \rho\cdot(\mathsf{aux}_{C,j})_{j\in K} + \rho^2\cdot(\mathsf{aux}_{R,j})_{j\in K},$$

$\mathsf{Sim}(s,(N,n),(\mathsf{c_t},\mathsf{c_f}),[\mathsf{T}(s)]_1)$:

1. $\rho_m \leftarrow\$ \mathbb{F}$; $m(s) \leftarrow \rho_m \nu_{\mathbb{K}}(s)$;
2. $R_S, \rho_S \leftarrow\$ \mathbb{F}$; $S(X) \leftarrow XR_S(X) + \rho_S \nu_{\mathbb{K}}(X)$;
3. Send $[m(s), S(s)]_1$;

---

4. Obtain β;
5. $\rho_A \leftarrow\$ \mathbb{F}$; $A(s) \leftarrow \rho_A \nu_{\mathbb{K}}(s)$;
6. $\rho_B(X) \leftarrow\$ \mathbb{F}_{\leq 1}[X]$; $B(s) \leftarrow \rho_B(s)\nu_{\mathbb{K}}(s)$;
7. $[Q_B(s)]_1 \leftarrow (B(s)(\mathsf{c_f} + \beta[1]_1) - 1)/\nu_{\mathbb{H}}(s)$;
8. Send $[A(s), B(s), Q_B(s)]_1$;

---

9. Obtain γ, η;
10. $[D(s)]_1 \leftarrow B(\gamma)(\mathsf{c_f} + \beta[1]_1) - 1 - \nu_{\mathbb{H}}(\gamma) \cdot [Q_B(s)]_1$;
11. // Note that $D(s) = \left(B(\gamma) - \frac{B(s)\nu_{\mathbb{H}}(\gamma)}{\nu_{\mathbb{H}}(s)}\right)(\mathsf{F}(s) + \beta) - 1 + \frac{\nu_{\mathbb{H}}(\gamma)}{\nu_{\mathbb{H}}(s)}$;
    // Replacing s with γ, we get that $D(\gamma) = 0$;
    Claim $D_\gamma \leftarrow 0$;
    Claim $B_\gamma \leftarrow B(\gamma) = \rho_B(\gamma)\nu_{\mathbb{K}}(\gamma)$;
12. $[P(s)]_1 \leftarrow \frac{1}{s-\gamma}((B(s) - B(\gamma))[1]_1 + \eta[D(s)]_1)$;
13. $R_C^*(s) \leftarrow \eta \cdot R_S \cdot U(s)$;
14. Choose $Q(s)$ that makes the verifier accept:
    $[Q(s)]_1 \leftarrow \rho_A(s)(\beta + \eta + [\mathsf{T}(s)]_1) - \frac{\eta}{\vartheta}\rho_B(s) - \rho_m(s) + \eta^2\rho_S(s)$;
15. Send $\left[P(s), R_C^*(s), Q(s)\right]_1, B_\gamma$.

Figure A.1: Simulator of cq^+

which, thanks to the $\mathbb{F}$-linearity of CP.Preproc, are valid auxiliary information for $(\bar{\mathbf{t}}^*, \bar{\mathbf{f}}^*)$ and then it runs the prover of CP whose running time is $\mathsf{poly}(n \cdot d, n)$.

Completeness is almost straightforward. Let $K = \{k_1, \ldots, k_n\}$ be such that $\mathbf{T}_{|K} = \mathbf{F}$ notice that for any i and for any j we have $\mathbf{T}_{k_i,j} + \rho j + \rho^2 k_i = \mathbf{F}_{i,j} + \rho j + \rho^2 \mathbf{S}_{i,j}$. In fact, $\mathbf{T}_{k_i,j} = \mathbf{F}_{i,j}$ by the hypothesis for completeness and $\mathbf{S}_{i,j} = k_i$ by inspection of the prover's algorithm. Moreover, we have that $\sigma(\omega X) - \sigma(X)$ evaluated in ω^{di+j} is equal to $\mathbf{S}_{i,j+1} - \mathbf{S}_{i,j} = 0$ for $j \in \{1, d-1\}$ and $\mathbf{S}_{i+1,0} - \mathbf{S}_{i,d}$ otherwise. Thus the polynomial $\sigma(\omega X) - \sigma(X)$ is divisible by $\nu_{\tilde{\mathbb{H}}}(X)$.

A

Derive(srs, N, n)**:** // Assume that $n \mid N$, $\mathsf{srs} = (\left[(s^j)_{j\in[N_1]}\right]_1, \left[(s^j)_{j\in[N_2]}\right]_2)$ for any $N_1 \geq N+\max(\mathsf{b}_F, 1)-1$ and $N_2 \geq N+\max(\mathsf{b}_F, 1)+1$.

Set $\mu = N_1 - N + 2$; define $U(X) := (X^\mu - 1)$, $\vartheta = N/n$, and $\mathsf{z}(X) = \nu_{\mathbb{K}\setminus\mathbb{H}}(X)$;

Let $\left\{r_j^{\mathbb{K}}(X) = \frac{\lambda_j^{\mathbb{K}}(X) - \lambda_j^{\mathbb{K}}(0)}{X} U(X)\right\}_{j\in[N]}$, and $\left\{r_i^{\mathbb{H}}(X) = \frac{\lambda_i^{\mathbb{H}}(X)\mathsf{z}(X) - \lambda_i^{\mathbb{H}}(0)}{X} U(X)\right\}_{i\in[n]}$;

Compute $\mathsf{ek}_{N,n} := \left[(r_j^{\mathbb{K}}(s))_{j=1}^N, (r_i^{\mathbb{H}}(s))_{i=1}^n, U(s), \nu_{\mathbb{K}}(s), s\nu_{\mathbb{K}}(s), (x^i)_{i=0}^{n+\mathsf{b}_F-1}\right]_1$

Compute $\mathsf{vk}_{N,n} := [1, U(s), \mathsf{z}(s)U(s), \nu_{\mathbb{K}}(s)U(s)]_2$

Return $(\mathsf{ek}_{N,n}, \mathsf{vk}_{N,n})$.

Preproc$(\mathsf{srs}, \mathbf{t}, \rho_{\mathsf{T}})$**:**

Define $\mathsf{T}(X) := \sum_{j=1}^N = \mathbf{t}_j \lambda_j^{\mathbb{K}}(X) + \rho_{\mathsf{T}} \cdot \nu_{\mathbb{K}}(X)$, and compute $\tilde{\mathsf{c}}_{\mathbf{t}} \leftarrow [\mathsf{T}(s)U(s)]_2$;

Compute $\mathsf{aux}_{\mathbf{t}} := (\mathbf{t}, \left[(Q_j(s))_{j=1}^N, \mathsf{T}(s)\right]_1)$;

Return $(\mathsf{aux}_{\mathbf{t}}, \tilde{\mathsf{c}}_{\mathbf{t}})$.

Prove$(\mathsf{ek}_{N,n}, (\mathsf{c}_{\mathbf{t}}, \mathsf{c}_{\mathbf{f}}), \mathsf{aux}_{\mathbf{t}}, (\mathbf{f}, \rho_{\mathsf{F}}(X)))$**:** //$\mathsf{c}_{\mathbf{f}} = \left[\sum_i \mathbf{f}_i \lambda_i^{\mathbb{H}}(s) + \rho_{\mathsf{F}}(s)\nu_{\mathbb{H}}(s)\right]_1$, $deg(\rho_{\mathsf{F}}) = \mathsf{b}_F$.

Compute $\mathbf{m} = (m_1, \ldots, m_N)$ s.t. $\forall j : \mathbf{t}_j$ appears m_j times in $\mathbf{f}$; samples $\rho_m \leftarrow\$ \mathbb{F}$;

Compute $[m(s)]_1 \leftarrow \sum_{j=1}^N m_j \cdot \left[\lambda_j^{\mathbb{K}}(s)\right]_1 + \rho_m \cdot [\nu_{\mathbb{K}}(s)]_1$; // n scalar mults

Sample $R_S, \rho_S \leftarrow\$ \mathbb{F}$ and compute $[S(s)]_1 \leftarrow R_S \cdot s + \rho_S \cdot \nu_{\mathbb{K}}(s)$;

$\beta \leftarrow \mathsf{RO}(\mathsf{vk}_{N,n} \| (\mathsf{c}_{\mathbf{t}}, \mathsf{c}_{\mathbf{f}}) \| [m(s)]_1)$ //Fiat-Shamir challenge.

Sample $\rho_A \leftarrow\$ \mathbb{F}$, $\rho_B(X) \leftarrow\$ \mathbb{F}_{\leq 1}[X]$;

Let $A_j \leftarrow m_j/(\mathbf{t}_j + \beta)\ \forall j = 1, \ldots, N$ and $B_i \leftarrow 1/(\mathbf{f}_i + \beta)\ \forall i = 1, \ldots, n$;

Compute $[A(s)]_1 \leftarrow \sum_{j=1}^N A_j \left[\lambda_j^{\mathbb{K}}(s)\right]_1 + \rho_A \cdot [\nu_{\mathbb{K}}(s)]_1$;

Compute $[B(s)]_1 \leftarrow \sum_{i=1}^n B_i \left[\lambda_i^{\mathbb{H}}(s)\right]_1 + \rho_B(s) \cdot [\nu_{\mathbb{H}}(s)]_1$;

Compute $Q_B(X) \leftarrow (B(X)(\mathsf{F}(X) + \beta) - 1)/\nu_{\mathbb{H}}(X)$ and $[Q_B(s)]_1$;

$(\gamma, \eta) \leftarrow \mathsf{RO}(\beta \| [A(s), B(s), Q_B(s), S(s)]_1)$;//Fiat-Shamir challenge.

Compute $B_\gamma \leftarrow B(\gamma)$, $D(X) \leftarrow B_\gamma \cdot (\mathsf{F}(X) + \beta) - 1 - Q_B(X)\nu_{\mathbb{H}}(\gamma)$;

Compute $P(X) \leftarrow ((B(X) - B(\gamma)) + \eta D(X))/(X - \gamma)$ and $[P(s)]_1$;

Compute $\left[R_C^*(s)\right]_1 \leftarrow \sum_{m_j \neq 0} A_j \cdot \left[r_j^{\mathbb{K}}(s)\right]_1 - \vartheta^{-1} \sum_{i=1}^n B_i \cdot \left[r_i^{\mathbb{H}}(s)\right]_1 + \eta^2 R_S \cdot [U(s)]_1$

Compute $[Q_A(s)]_1 \leftarrow \sum_{m_j \neq 0} A_j \cdot \left[Q_j(s)\right]_1 + \left[\rho_A(T(s) + \beta) - \rho_m\right]_1$;

Compute $[Q_C(s)]_1 \leftarrow \left[\rho_A + \vartheta^{-1}\rho_B(s)\right]_1$;

Compute $[Q(s)]_1 \leftarrow [Q_A(s)]_1 + \eta [Q_C(s)]_1 + \eta^2 \left[\rho_S\right]_1$;

Return $\pi = (\tilde{\mathsf{c}}_{\mathbf{t}}, \left[m(s), S(s), A(s), B(s), Q_B(s), P(s), R_C^*(s), Q(s)\right]_1, B_\gamma)$.

Verify$(\mathsf{vk}_{N,n}, (\mathsf{c}_{\mathbf{t}}, \mathsf{c}_{\mathbf{f}}), \pi)$**:**

Compute $[D(s)]_1 \leftarrow B_\gamma(\mathsf{c}_{\mathbf{f}} + \left[\beta\right]_1) - [1]_1 - \nu_{\mathbb{H}}(\gamma)\,[Q_B(s)]_1$.

Return 1 if and only if the following holds:

(i) $e([A(s)]_1, \mathsf{c}_{\mathbf{t}}) \cdot e((\beta+\eta) \cdot [A(s)]_1 - [m(s)]_1 + \eta^2 [S(s)]_1, [U(s)]_2) \cdot e(\eta/\vartheta \cdot [B(s)]_1, [\mathsf{z}(s)U(s)]_2)^{-1} \cdot e([Q(s)]_1, [\nu_{\mathbb{K}}(s)U(s)]_2)^{-1} = e(\eta \cdot \left[R_C^*(s)\right]_1, [x]_2)$,

(ii) $e([B(s)]_1 + \eta\,[D(s)]_1 - \left[B_\gamma\right]_1, [1]_2) = e([P(s)]_1, \left[s - \gamma\right]_2)$

(iii) $e([1]_1, \tilde{\mathsf{c}}_{\mathbf{t}}) = e(\mathsf{c}_{\mathbf{t}}, [U(s)]_2)$

Figure A.2: Our fully zero-knowledge lookup argument zkcq^+.

As for zero-knowledge, by randomizing σ by summing $\nu_{\mathbb{K}}(X) \cdot r(X)$ where r is a random polynomial with $\deg(r) = 2$, we have that the values $\sigma(s), \sigma(\zeta), \sigma(\omega \cdot s)$ are uniformly distributed. In particular, one could sample random value z and random group elements

for $\mathsf{c}_{R,n}, \mathsf{c}_{R',n}$ and use the zero-knowledge simulator of the proof of evaluation and the zero-knowledge simulator of CP.

We focus on knowledge soundness in the AGM. Notice that the prover and verifier algorithm define a one-round public-coin protocol where we applied the Fiat-Shamir transform.

The adversary outputs valid representations for the proof elements that we can parse as polynomials. Let $\tilde{\sigma}_1(X)$ and $\tilde{\sigma}_2(X)$ be the polynomials underlying the commitments $\mathsf{c}_{R,n}$ and $\mathsf{c}_{R',n}$ notice that by the verification equations in items (i) and (ii) there exist two polynomials W_1 and W_2 such that:

$$\tilde{\sigma}_1(X) - z = W_1(X)(X - \omega \cdot \zeta) \tag{A.1}$$

$$\tilde{\sigma}_2(X) - z = W_2(X)(X - \zeta) \tag{A.2}$$

By change of variable we have $\tilde{\sigma}_1(\omega X) - z = W_1(\omega X)(\omega \cdot X - \omega \cdot \zeta) = \omega W_1(\omega X)(X - \zeta)$, and thus $\sigma_1(\omega X) - \tilde{\sigma}_2(X)$ is divisible by $(X - \zeta)$. Since ζ is sampled uniformly at random after $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ are defined, by the Swartz-Zippel Lemma we have that $\tilde{\sigma}_1(\omega X) = \tilde{\sigma}_2(X)$.

By the pairing equation in item (iii) we have

$$\sigma_1(\omega X) - \sigma_1(X) = W_3(X) v_{\bar{\mathbb{H}}}(X)$$

and thus for any i, j with $j \in [1, d-1]$ we have $\sigma_1(\omega^{d \cdot i + j + 1}) = \sigma_1(\omega^{d \cdot i + j})$. The latter implies that there exists a multi-set[2] of indexes $K = \{k_1, \ldots, k_n\}$ such that $\sigma_1(\omega^{d \cdot i + j}) = k_i$.

Finally, consider the vectors $\hat{\mathbf{t}}, \hat{\mathbf{f}}$ with elements in $\mathbb{F}^3$ such that $\hat{\mathbf{t}}_{d \cdot i + j} = (\mathbf{T}_{i,j}, \mathbf{C}^{(N)}_{i,j}, \mathbf{R}_{i,j})$ and $\hat{\mathbf{f}}_{d \cdot i + j} = (\mathbf{F}_{i,j}, \mathbf{C}^{(n)}_{i,j}, \sigma_1(\omega^{d \cdot i + j}))$. Notice that $\hat{\mathbf{f}} \prec \hat{\mathbf{t}}$ if and only if $\mathbf{T}_{|K} = \mathbf{F}$. Moreover, the Swartz-Zippel Lemma implies that the family of hash functions with signature $\mathbb{F}^3 \to \mathbb{F}$ that maps (x_0, x_1, x_2) to $\sum \rho^i x_i$ with key ρ sampled uniformly at random over $\mathbb{F}$ form a universal hash function family, namely for any $\mathbf{x}, \mathbf{x}'$ such that $\mathbf{x} \neq \mathbf{x}'$ $\Pr[h(\mathbf{x}) = h(\mathbf{x}')] = 1/q$. Notice that, the vectors $\hat{\mathbf{t}}$ and $\hat{\mathbf{f}}$ are fully defined before ρ is sampled. Thus by union bound over all the tuples of coordinates of $\hat{\mathbf{f}}$ and $\hat{\mathbf{t}}$ and by the universal hash property, we have that with $1 - \Omega(nd/q)$ probability $\bar{\mathbf{f}}^* \prec \bar{\mathbf{t}}^*$ implies that $\hat{\mathbf{f}} \prec \hat{\mathbf{t}}$. □

A.4.1. ROWS-COLUMNS MATRIX LOOKUP

In this section we consider a more general notion of the sub-matrix relation, to which we refer as (rows-columns) sub-matrix relation, where $\mathbf{F} \prec \mathbf{T}$ with $\mathbf{F} \in \mathbb{F}^{n \times d}$, $\mathbf{T} \in \mathbb{F}^{N \times D}$ and $N, D, n, d \in \mathbb{N}$ holds true if and only if there exist (multi)sets $R = \{r_1, \ldots, r_n\}$ and $C = \{c_1, \ldots, c_d\}$ with $\mathbf{F}_{i,j} = \mathbf{T}_{r_i, c_j}$ for any i, j. Similar to the notion of rows sub-matrix, we define $\mathbf{T}_{|R \times C}$ be the sub-matrix of $\mathbf{T}$ such that $(\mathbf{T}_{|R \times C})_{i,j} = \mathbf{T}_{r_i, c_j}$ for any i, j. We realize an argument system for the relation:

$$\hat{\mathcal{R}}'_{\mathsf{zklookup}} = \left\{ \mathsf{pp}; (N, D, n, d); \varepsilon; \mathbf{T}, \mathbf{F} : \begin{array}{c} \mathbf{F} \prec \mathbf{T} \\ |\mathbf{T}_j| = N \times D, |\mathbf{F}| = n \times d \end{array} \right\}$$

Notice that we are not zero-knowledge neither with respect of the number of columns nor the number of rows of the sub-matrix $\mathbf{F}$.

[2] K is a multiset because there might exist i, k such that $k_i = k_j$.

In fig. A.3 we describe our second scheme $\mathsf{mtx}^*[\mathsf{CP}]$ for rows-columns matrix lookup, namely a matrix lookup argument w.r.t. the more general relation described above, the that runs internally a lookup argument CP for KZG-based vector commitment scheme. In the description of the scheme, we let $\mathbb{K}$ (resp. $\mathbb{H}$) be a multiplicative subgroup of $\mathbb{F}$ of order $N \cdot D$ (resp. of order $n \cdot d$), we let $\omega := \omega_{n \cdot d}$ be the fixed generator for $\mathbb{H}$ and we consider the following matrices and polynomial:

1. the matrix $\mathbf{R} \in \mathbb{F}^{N \times d}$ where $R_{i,j} = i$,
2. for any k the matrix $\mathbf{C}^{(k)} \in \mathbb{F}^{k \times d}$ where $C_{i,j} = j$.
3. Let $v_R(X)$ be the vanishing polynomial of the set $\mathbb{H}_R = \{\omega^{d \cdot i + j} : j \in [1, d-1], i \in [n]\}$.
4. Let $v_{\mathbb{H}}(X)$ be the vanishing polynomial of the multiplicative subgroup $\mathbb{H}$.

Theorem A.4. *The lookup argument* $\mathsf{mtx}^*[\mathsf{CP}]$ *defined in fig. A.3 is knowledge-sound in the AGM and ROM under the* $(N \cdot d, N \cdot d)$*-PDL assumption and assuming that* CP *is knowledge-sound. Furthermore, the protocol is zero-knowledge assuming* CP *is zero-knowledge.*

The proof of the theorem is very similar to the proof of theorem 6.4, the only difference is that the prover additionally shows that $\sigma^C(X)$ has a well-defined tensor-product structure that we prove in the next lemma.

Lemma A.5. *Let* $\mathbb{H} = \langle \omega \rangle$ *be a multiplicative subgroup of* $\mathbb{F}$ *of order* $n \cdot d$ *with* $d \geq 1$*, there exists* $\mathbf{c}$ *of length* d *such that* $\sigma^C(X)$ *is the LDE of* $\mathbf{c} \otimes \mathbf{1}$ *(over* $\mathbb{H}$*) if and only if* $\sigma^C(\omega^d \cdot X) - \sigma^C(X) \equiv 0 \mod v_{\mathbb{H}}(X)$.

Proof. The first implication is easy. In fact, let $\sigma^C(X)$ be the LDE of $\mathbf{c} \otimes \mathbf{1}$ then $\sigma^C(\omega^{(i+1) \cdot d + j \mod nd}) = \sigma^C(\omega^{i \cdot d + j \mod nd})$ for any i and $j \in [d]$. We can prove the other direction by induction.

Let $v_i(X)$ be the vanishing polynomial of the set $\mathbb{H}_i = \{\omega^j : 0 \leq j < i \cdot d\}$, we can show that if $\bar{\sigma}(X) := \sigma^C(\omega^d \cdot X) - \sigma^C(X) \equiv 0 \mod v_i(X)$ and $i \leq n$ then $\exists \mathbf{c}, \mathbf{d}$ such that $\omega^C(X)$ is the LDE of $(\mathbf{c} \otimes \mathbf{1}_i \| \mathbf{d})$ and $\mathbf{1}_i$ has length i.

- For $i = 1$ the statement is trivially true because $\forall j \in [d] : \sigma^C(\omega^{d+j}) = \sigma^C(\omega^j)$ thus there exists $\mathbf{c}, \mathbf{d}$ such that σ^C is the LDE of $\mathbf{c} \| \mathbf{c} \| \mathbf{d}$.
- For $n \geq i > 1$ we have that $\bar{\sigma}(X) \equiv 0 \mod v_i(X)$ implies $\bar{\sigma}(X) \equiv 0 \mod v_{i-1}(X)$, thus σ^C is the LDE of $\mathbf{c} \otimes \mathbf{1}_{i-1} \| \mathbf{d}$. We need to show that $\mathbf{d} = \mathbf{c} \| \mathbf{d}'$ for some vector $\mathbf{d}'$. Notice that $\sigma^C(\omega^d \cdot \omega^{(i-1) \cdot d + j}) - \sigma^C(\omega^{(i-1) \cdot d + j}) = 0$ for $j \in [d]$ thus, if $i < n$, the first d coordinates of $\mathbf{d}$ agree with the last d coordinate of $\mathbf{c} \otimes \mathbf{1}_{i-1}$, which means that $\mathbf{d} = \mathbf{c} \| \mathbf{d}'$ for some $\mathbf{d}'$, if $i = n$ the first d coordinates of $\mathbf{d}$ agree with the first d coordinates of $\mathbf{c} \otimes \mathbf{1}_{n-1}$, which means that $\mathbf{d} = \mathbf{c}$.

□

Derive(srs, N, d, n)**:**

Let $\bar{\mathbf{t}}, \bar{\mathbf{f}}, \bar{\mathbf{r}}_N, \bar{\mathbf{c}}_N$ and $\bar{\mathbf{c}}_n$ be vectorizations of the matrices $\mathbf{T}, \mathbf{F}, \mathbf{R}, \mathbf{C}^{(N)}$ and $\mathbf{C}^{(n)}$.
Compute $\mathsf{c}_{r,N} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{r}}_N)$ and $\mathsf{c}_{c,N} \leftarrow \mathsf{Com}(\mathsf{ck}, \bar{\mathbf{c}}_N)$.
Compute $(\mathsf{ek}', \mathsf{vk}') \leftarrow \mathsf{CP.Derive}(\mathsf{srs}, Nd, nd)$.
Return $(\mathsf{ek}', \mathsf{vk}_n)$ where $\mathsf{vk}_n = (\mathsf{c}_{r,N}, \mathsf{c}_{c,N}, [\nu_R(s)]_2, \mathsf{vk}')$.

Preproc$(\mathsf{srs}, \mathbf{T})$**:**

Compute $(\mathsf{aux}_{T,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{t}})$,
$(\mathsf{aux}_{R,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{r}}_N)$,
$(\mathsf{aux}_{C,j})_{j\in[Nd]} \leftarrow \mathsf{CP.Preproc}(\mathsf{srs}, \bar{\mathbf{c}}_N)$.
Let $\mathsf{aux}_i = (\mathsf{aux}_{T,di+j}, \mathsf{aux}_{R,di+j}, \mathsf{aux}_{C,di+j})_{j\in[d]}$.
Return $(\mathsf{aux}_i)_{i\in[N]}$.

Prove$(\mathsf{ek}, (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}), \mathbf{F}, (\mathsf{aux}_j)_{j\in K})$**:** //$\mathbf{T}_{|R\times C} = \mathbf{F}$, $R = \{r_1, \ldots, r_n\}$, $C = \{c_1, \ldots, c_d\}$.

Let $\mathbf{S}^R$ be s.t. $\mathbf{S}^R_{i,j} = r_i$ for $i \in [n]$, $j \in [d]$. Let $\mathbf{S}^C$ be s.t $\mathbf{S}^C_{i,j} = c_j$ for $i \in [n]$, $j \in [d]$.
Let $\sigma^R(X)$ (resp. $\sigma^C(X)$) be a randomized LDE over $\mathbb{H}$ of the vectorization of $\mathbf{S}^R$ (resp. $\mathbf{S}^C$).
Compute $w^R(X)$ such that $\sigma^R(\omega \cdot X) - \sigma^R(X) = w^R(X) \cdot \nu_R(X)$.
Compute $w^C(X)$ such that $\sigma^C(\omega^d \cdot X) - \sigma^C(X) = w^C(X) \cdot \nu_{\mathbb{H}}(X)$.
$(\rho, \zeta) \leftarrow \mathsf{RO}(\mathsf{vk}_n \| (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}) \| (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w))$. //Fiat-Shamir challenge.
Compute $z^R \leftarrow \sigma(\omega \cdot \zeta), z^C \leftarrow \sigma(\omega^d \cdot \zeta)$.
Compute proofs π_R and $\pi_{R'}$ for $\hat{\mathcal{R}}_{\mathsf{eval}}(\omega \cdot \zeta, z^R; \sigma^R(X)) = 1$ and $\hat{\mathcal{R}}_{\mathsf{eval}}(\zeta, z^R; \sigma(\omega \cdot X)) = 1$;
Compute proofs π_C and $\pi_{C'}$ for $\hat{\mathcal{R}}_{\mathsf{eval}}(\omega^d \cdot \zeta, z^C; \sigma^C(X)) = 1$ and $\hat{\mathcal{R}}_{\mathsf{eval}}(\zeta, z^C; \sigma(\omega^d \cdot X)) = 1$;
Let π^* proof for $\hat{\mathcal{R}}_{\mathsf{zklookup}}((N \cdot d, n \cdot d); \varepsilon; (\bar{\mathbf{t}}^*, \bar{\mathbf{f}}^*)) = 1$ where

$$\bar{\mathbf{t}}^* = \bar{\mathbf{t}} + \rho \cdot \bar{\mathbf{c}}_N + \rho^2 \cdot \bar{\mathbf{r}}_N \qquad \bar{\mathbf{f}}^* = \bar{\mathbf{f}} + \rho \cdot \bar{\boldsymbol{\sigma}}^C + \rho^2 \cdot \bar{\boldsymbol{\sigma}}^R \tag{A.3}$$

Return $\pi = ([\sigma(s)]_1, [\sigma(\omega \cdot s)]_1, [w(s)]_1, \pi_R, \pi_{R'}, \pi^*, z)$.

Verify$(\mathsf{vk}_n, (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}), \pi)$**:**

Parse the proof $\pi = (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w, \pi_R, \pi_{R'}, \pi^*, z)$.
$(\rho, \zeta) \leftarrow \mathsf{RO}(\mathsf{vk}_n \| (\mathsf{c}_\mathbf{T}, \mathsf{c}_\mathbf{F}) \| (\mathsf{c}_{R,n}, \mathsf{c}_{R',n}, \mathsf{c}_w))$. //Fiat-Shamir challenge.
Compute $\mathsf{c}^*_\mathbf{T} \leftarrow \mathsf{c}_\mathbf{T} + \rho \mathsf{c}_{c,N} + \rho^2 \mathsf{c}_{r,N}$ and $\mathsf{c}^*_\mathbf{F} \leftarrow \mathsf{c}_\mathbf{F} + \rho \mathsf{c}_{c,n} + \rho^2 \mathsf{c}_{R,n}$.
Return 1 if the following checks hold (else 0):

(i) $\mathsf{Verify}_{\mathsf{eval}}(\mathsf{ck}, (\mathsf{c}_{R,n}, \omega \cdot \zeta, z)) = 1$,
(ii) $\mathsf{Verify}_{\mathsf{eval}}(\mathsf{ck}, (\mathsf{c}_{R',n}, \zeta, z)) = 1$,
(iii) $e(\mathsf{c}_{R',n} - \mathsf{c}_{R,n}, [1]_2) = e(\mathsf{c}_w, [\nu_{\bar{\mathsf{k}}}(s)]_2)$,
(iv) $\mathsf{CP.Verify}(\mathsf{srs}, \mathsf{vk}', (\mathsf{c}^*_\mathbf{T}, \mathsf{c}^*_\mathbf{F}), \pi^*) = 1$.

Figure A.3: Our rows-columns Matrix Lookup Argument mtx^*[CP].

A

Procedure $\mathsf{T}(x)$:

$n \leftarrow 1$ // root node
while n is not a leaf **do**
  fetch $\mathbf{E}_n, \mathbf{T}_n$
  if $\forall j \in [d] : \mathbf{E}_{n,j} = 1 \Rightarrow x_j < \mathbf{T}_{n,j}$ **then** $n \leftarrow$ left child of n
  elseif $\forall j \in [d] : \mathbf{E}_{n,j} = 1 \Rightarrow x_j \geq \mathbf{T}_{n,j}$ **then** $n \leftarrow$ right child of n
  else return $\perp$
fetch v_n **from** $\mathbf{v}$ // Vector of all the labels
return v_n

Figure A.4: The pseudo-code of an evaluation of a quasi-complete decision tree.

A.5. ADDITIONAL MATERIAL ON SECTION 6.6

A.5.1. THE EXTENDED ENCODING OF DECISION TREES

Lemma A.6. *Let T be a quasi-complete decision tree with N_{tot} nodes and $(\mathbf{N}^-, \mathbf{N}^-)$ be a boxes-encoding of T. Let $\mathbf{v}$ be the vector of the labels assigned to the leaf nodes of T, namely for any $i \in [N_{\mathsf{int}}+1, N_{\mathsf{tot}}]$, we have v_i as the label assigned to the i-th leaf. For any $\mathbf{x} \in [B]^d$, $\mathsf{T}(\mathbf{x}) = v_{k(\mathbf{x})}$ or $\mathsf{T}(\mathbf{x}) = \perp$.*

Proof. Let $n_1, \dots, n_s$ be the nodes traversed by the computation of $\mathsf{T}(\mathbf{x})$, we prove that $\mathbf{x}$ is contained in $(\mathbf{N}^-_{n_j}, \mathbf{N}^-_{n_j})$ for any j or $T(\mathbf{x}) = \perp$. Notice that n_1 is the root, namely $n_1 = 1$, and by (1) of definition 6.7 we clearly have $\mathbf{x}$ is contained in $(\mathbf{N}^-_1, \mathbf{N}^-_1)$. Moreover, assume that at the i-th step, $\mathbf{x}$ is contained in $(\mathbf{N}^-_{n_i}, \mathbf{N}^-_{n_i})$. If $\forall j \in [d] : \mathbf{E}_{n_i,j} = 1 \Rightarrow x_j < \mathbf{T}_{n_i,j}$ then n_{i+1} is the left child of n_i and because of eqs. (6.14) and (6.15) we have that $\mathbf{x}$ is contained in $(\mathbf{N}^-_{n_{i+1}}, \mathbf{N}^-_{n_{i+1}})$. Similarly, if $\forall j \in [d] : \mathbf{E}_{n_i,j} = 1 \Rightarrow x_e \geq \mathbf{T}_{n_i,j}$, then n_{i+1} is the right child of n_i and because of eqs. (6.14) and (6.16) we have that $\mathbf{x}$ is contained in $(\mathbf{N}^-_{n_{i+1}}, \mathbf{N}^-_{n_{i+1}})$. Otherwise, we have $\mathsf{T}(\mathbf{x}) = \perp$.

Because of eqs. (6.15) and (6.16) the boxes of the left and right children are disjoint (namely, there isn't any $\mathbf{x}$ that is contained in both boxes). Thus by induction on the structure of the tree, the set of the boxes of all the leaves are pair-wise disjointed. This implies that if $\mathsf{T}(\mathbf{x}) \neq \perp$ then $k(\mathbf{x})$ is uniquely defined. Thus n_s is equal to $k(\mathbf{x})$. □

Lemma A.7. *Consider a tuple $(\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$ such that the following constraints hold:*

a) The following equations hold:

$$\mathbf{N}^-_1 = \mathbf{0}, \mathbf{N}^-_1 = \mathbf{B} + \mathbf{1}, \tag{6.17}$$

$$\mathbf{L} \cdot \mathbf{N}^- = \mathbf{P}^-, \mathbf{R} \cdot \mathbf{N}^- = \mathbf{P}^-, \tag{6.18}$$

$$\mathbf{E} \circ (\mathbf{L} \cdot \mathbf{N}^- - \mathbf{R} \cdot \mathbf{N}^-) = \mathbf{0} \tag{6.19}$$

$$(\mathbf{1} - \mathbf{E}) \circ (\mathbf{P}^- - \mathbf{R} \cdot \mathbf{N}^-) = \mathbf{0}, \qquad (\mathbf{1} - \mathbf{E}) \circ (\mathbf{P}^- - \mathbf{L} \cdot \mathbf{N}^-) = \mathbf{0} \tag{6.20}$$

b) All the boxes are not empty. Namely, for all i, j we have $\mathbf{N}^-_{i,j} < \mathbf{N}^-_{i,j}$.

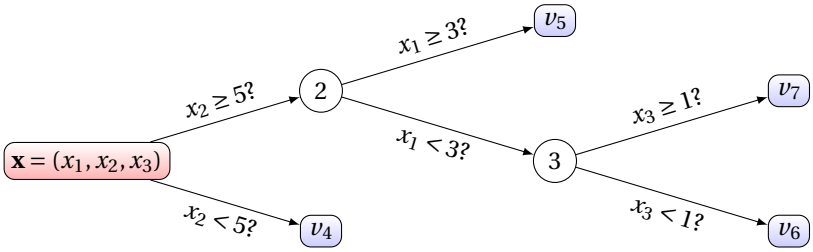


Figure A.5: Example of decision tree ($d = 3, N_{\text{int}} = 3, N_{\text{tot}} = 7$). $\mathbf{x}$ is the input to the tree. Internal nodes are marked by their (circled) index only, otherwise the subscript in the labels (v_i-s) marks their index. The root is implicitly indexed as node 1.

$$\mathbf{L} = \overbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}^{N_{\text{tot}}} \Big\} N_{\text{int}} \quad \mathbf{R} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{N}^- = \overbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \\ 3 & 5 & 0 \\ 0 & 5 & 0 \\ 0 & 5 & 1 \end{bmatrix}}^{d} \Big\} N_{\text{tot}} \quad \mathbf{N}^- = \begin{bmatrix} B+1 & B+1 & B+1 \\ B+1 & B+1 & B+1 \\ 3 & B+1 & B+1 \\ B+1 & 5 & B+1 \\ B+1 & B+1 & B+1 \\ 3 & B+1 & 1 \\ 3 & B+1 & B+1 \end{bmatrix} \quad \mathbf{E} = \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{d} \Big\} N_{\text{int}}$$

Figure A.6: Matrix examples for decision tree in fig. A.5. Notice that the boundaries in $\mathbf{N}^-$ and $\mathbf{N}^-$ consist of a half-open interval that is greater or equal to $\mathbf{N}^-$ and less than $\mathbf{N}^-$ (e.g. $0 \leq x_1 < 3$ for node 3). Notice how the Hadamard product of a row of $\mathbf{1} - \mathbf{E}$ can be used to show two sibling boxes are the same except in one coordinate, e.g. for nodes 6 and 7 we use the third row of $\mathbf{1} - \mathbf{E}$ (node 3 is their parent) to show that it should hold $(\mathbf{1} - \mathbf{E})_3 \circ (\mathbf{N}^-_6 - \mathbf{N}^-_7) = \mathbf{0}$ and $(\mathbf{1} - \mathbf{E})_3 \circ (\mathbf{N}^-_6 - \mathbf{N}^-_7) = \mathbf{0}$. Also, for the split coordinate, the right bound of the left child should equal the left bound of the right child and equal the threshold of their parent. For example, for node 2, the right bound of the left child (node 3) for x_1 is $(\mathbf{E}_2 \circ (\mathbf{L} \cdot \mathbf{N}^-)_2)_1 = 3$. Similarly, for the right child, we have $(\mathbf{E}_2 \circ (\mathbf{R} \cdot \mathbf{N}^-)_2)_1 = 3$. We have the threshold for node 2 and x_1 is $\mathbf{T}_{2,1} = 3$.

c) The matrix $\begin{pmatrix} \mathbf{L} \\ \mathbf{R} \end{pmatrix}$ *is a (row) permutation of the (squared) matrix* $(\mathbf{0} \| \mathbf{I}_{N_{\text{tot}}-1})$ *(the matrix whose rows are the row vectors* $(\mathbf{e}_i)_{i \in [2, N_{\text{tot}}]}$ *of length* N_{tot}*).*

Then there exists a quasi-complete decision tree T *with* N_{tot} *nodes such that* $\mathsf{Encode}(\mathsf{T}) = (\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$.

Proof. First, notice that the constraint in eq. (6.17) is necessary by definition of box-encoding of a tree.

From item c), for any $p \in [N_{\mathsf{int}}]$, $\mathbf{L}_p$ and $\mathbf{R}_p$ are elementary vectors, so there exists unique l, r such that $\mathbf{L}_{i,l} = 1, \mathbf{R}_{i,r} = 1$. Since $N_{\mathsf{tot}} - 1 = 2N_{\mathsf{int}}$ and $\mathcal{L} \cup \mathcal{R} = [2, N_{\mathsf{tot}}]$, the rows of the matrix $\frac{\mathbf{L}}{\mathbf{R}}$ are linearly independent. As a result, all internal nodes have one left child and one right child.

We define a procedure that, upon the input data structure $(\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$ such that the hypothesis of the lemma holds, computes an (alleged) quasi-complete decision tree. We then show that the latter is indeed a quasi-complete decision tree, namely, that (I) the resulting (indirect) graph is acyclic and the number of edges in the graph is $N_{\mathsf{tot}} - 1$ (thus, it is a tree), moreover, the out-degree of any of the nodes is either 2 or 0 (thus it is a binary tree), and (II) for each internal node p, the procedure defines the feature vector $\mathbf{E}_p \in \{0,1\}^d$ and threshold vector $\mathbf{T}_p \in [B]^d$.

- Start with a fully disconnected graph with N_{tot} nodes. For any $p \in [N_{\mathsf{int}}]$, let l, r be the unique indexes such that $\mathbf{L}_{p,l} = 1$ and $\mathbf{R}_{p,r} = 1$. Add the direct edge (p, l) and (p, r) to the direct graph.
- Set $\mathbf{T} \leftarrow \mathbf{E} \circ (\mathbf{L} \cdot \mathbf{N}^-)$. For any p, the p-th row of $\mathbf{T}$ and $\mathbf{E}$ are the feature and threshold vectors for p.
- Associate to the leaves the labels $\mathbf{v}$. Namely, for any $i > N_{\mathsf{int}}$, the i-th node gets assigned the label v_i.

We notice that for any p, $\mathbf{E}_p, \mathbf{T}_p$ are well defined. In fact

$$\mathbf{E}_p \circ (\mathbf{N}^-_l - \mathbf{T}_p) = \mathbf{E}_p \circ (\mathbf{N}^-_l - (\mathbf{L} \cdot \mathbf{N}^-)_p) = \mathbf{0}$$

where the first equation comes by definition of $\mathbf{T}_p$ and the second by definition of $\mathbf{L}$. Notice that, by hypothesis of the lemma, we have $\mathbf{T}_p = \mathbf{R} \cdot \mathbf{N}^-$. Thus, with the same derivation as above, we have $\mathbf{E}_p \circ (\mathbf{N}^-_r - \mathbf{T}_p) = 0$. By definition of $\mathbf{L}$ (resp. $\mathbf{R}$) and eq. (6.20) we readily derive that eq. (6.16) holds for $\mathbf{E}$. Thus we have then proved condition (II).

We can focus on proving condition (I). First notice, it is easy to check that the out-degree of any of the nodes is at most 2, by definition of the procedure described above. Notice that the number of edges added by the procedure is the added number of rows in $\mathbf{L}$ and $\mathbf{R}$, namely $N_{\mathsf{tot}} - 1 = 2N_{\mathsf{int}}$. Thus we need to prove that the procedure did not add twice the same edge. This could only happen if there is a node $p \in [N_{\mathsf{int}}]$ such that for the same child node i, $\mathbf{L}_{p,i} = \mathbf{R}_{p,i} = 1$. However, we have proved that all internal nodes have one left and one right child. Given that the number of elements in $(\mathbf{L}, \mathbf{R})$ is $N_{\mathsf{tot}} - 1$, we have that any child node i can not serve as a child node more than one time. In other words, for any child node i, we have the only $p \in [N_{\mathsf{int}}]$ such that $\mathbf{L}_{p,i} = 1$ **or** $\mathbf{R}_{p,i} = 1$.

Define $P_i := \sum_j |\mathbf{N}^-_{i,j} - \mathbf{N}^-_{i,j}|$ as the "potential" associated to the box $(\mathbf{N}^-_i, \mathbf{N}^-_i)$ over the integers. For any $p \in [N_{\mathsf{int}}]$ with $\mathbf{L}_{p,l} = 1$ and $\mathbf{R}_{p,r} = 1$, we have $P_p > P_l$ and $P_p > P_r$. For the former, namely $P_p > P_l$, notice that because of eqs. (6.18) to (6.20) and item b), for any e_i with all $\mathbf{E}_{p,e_i} = 1$, we have it holds that $\mathbf{N}^-_{p,e_i} = \mathbf{N}^-_{l,e_i} < \mathbf{N}^-_{l,e_i} < \mathbf{N}^-_{r,e_i} = \mathbf{N}^-_{p,e_i}$, and thus $\mathbf{N}^-_{p,e_i} - \mathbf{N}^-_{p,e_i} > \mathbf{N}^-_{l,e_i} - \mathbf{N}^-_{l,e_i}$ while, $\mathbf{N}^-_{p,e_j} - \mathbf{N}^-_{p,e_j} = \mathbf{N}^-_{l,e_j} - \mathbf{N}^-_{l,e_j}$ for the other indexes e_j with all $\mathbf{E}_{p,e_j} = 0$. The latter, namely that $P_p > P_r$, follows similarly. Assume there

exists a cycle in (the indirect generalization of) the graph produced by the procedure described above, and let $(j_1, \ldots, j_k = j_1)$ be such a cycle. First, notice that because of the constraint in item c), every node has an in-degree at most 1 in the direct graph. Thus, if a cycle exists in the indirect graph, there is a cycle in the direct graph as well. Moreover, by construction, all the edges in the graph are of the form (p, l) or (p, r) for $p \in [N_{\mathsf{int}}]$. We have for any k that $P_{j_k} < P_{j_{k+1}}$ and thus $P_{j_1} < P_{j_1}$ which reaches a contradiction. Thus there are no cycles in the graph. □

A.5.2. EXTRACTABLE COMMITMENT TO DECISION TREES

Theorem 6.7. *The commitment scheme CS_{DT} defined in Figure 6.3 is hiding, and it is an extractable commitment scheme for the domain $\{\mathcal{T}^*_{N_{\mathsf{tot}},B,d}\}_{N_{\mathsf{tot}},d,B}$ in the AGM and assuming the building blocks are knowledge-sound and zero-knowledge.*

Sketch. We can prove hiding by relying on the hiding of the matrix commitment scheme and the zero-knowledge of the underlying CP-SNARKs[3].

By definition of extractable commitment we can interpret the commitment function as a first sub-procedure that generates a (binding) commitment and second procedure that generates a proof. By inspection of the algorithm, we can divide the commitment function above in this way. We need to prove knowledge soundness in the AGM for the derived CP-SNARK.

The extractor, using the algebraic representations, extract polynomials from the commitments $\mathsf{c}_{\_}, \mathsf{c}_{\_}, \mathsf{c}_v$, the commitments $\mathsf{c}'_{\_}, \mathsf{c}'_{\_}$, the sparse commitments $\mathsf{c}_L, \mathsf{c}_R$ and the commitment c_E. From such polynomials, the extractor can derive matrices $(\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$. In particular, the matrix $\mathbf{N}^-$ is defined as the sum of the (padded) matrices $\mathbf{F}^-$ extracted from $\mathsf{c}_{\_}$ and $\mathbf{P}^-$ extracted from $\mathsf{c}'_{\_}$ (similarly for $\mathbf{N}^-$). We show that the constraints of theorem 6.6 hold, thus the extractor can compute a valid quasi-complete decision tree T from the extracted extended encoding.

The validity of the proofs $\pi_{11}, \ldots, \pi_{14}$ enforce that $\mathbf{N}^-$ (resp. $\mathbf{N}^-$) stacks the matrix $\mathbf{P}^-$ on to of $\mathbf{F}^-$ (resp. $\mathbf{P}^-$ on top of $\mathbf{F}^-$). Moreover, the validity of the proofs π_{15}, π_{16} enforce the constraint eq. (6.17), as otherwise we would either break the knowledge soundness of $\mathsf{CP}_{\mathsf{sm}}$ or the binding property of the matrix commitment scheme.

The validity of the proofs π_1 and π_3 implies that eq. (6.18) indeed holds, as otherwise we would either break the knowledge soundness of $\mathsf{CP}_{\mathsf{lin}}$ or the binding property of the matrix commitment scheme.

The validity of the proof π_2 (resp. π_4) implies that the commitments c_{ln} (resp. c_{rn}) open to the matrix $\bar{\mathbf{L}} \cdot \mathbf{N}^-$ (resp. $\bar{\mathbf{R}} \cdot \mathbf{N}^-$), this coupled with the validity of the proof π_5 imply the constraints in eqs. (6.19) and (6.20). Again, we can formally prove this by a first reduction to the biding property of the matrix commitment (showing that the knowledge extractors for π_2, π_4 and π_5 should output the same matrices as computed by the algebraic representations), and then to the knowledge soundness of $\mathsf{CP}_{\mathsf{lin}}$ and $\mathsf{CP}_{\mathsf{had}}$.

The validity of the proof π_8 implies item b) in a straightforward manner. The validity of the proofs π_9 and π_{10} and the definition of the polynomial $id(X)$ by the KGen algo-

[3] Notice that, using higher degrees for the randomizers of the commitments, hiding would still hold even if the proofs leaked evaluation points (see the notion of leaky-zero-knowledge from [5]) from the commitments in π

rithm imply item c). In particular set $c'(X) \stackrel{\text{def}}{=} col_{\underline{\mathbf{L}}}(X) + col_{\underline{\mathbf{R}}}(X)$, we have $c'(h^i) = col_{\underline{\mathbf{L}}}(h^i)$ and $c'(h^{i+N_{\mathsf{int}}}) = col_{\underline{\mathbf{R}}}(h^i)$ for $i \in [N_{\mathsf{int}}]$. Moreover, $c'(X)$ is a permutation of $i(X)$ which, by definition, represents a sparse commitment of the matrix whose rows are the elementary vectors $(\mathbf{e}_{j+1})_{j\in[N_{\mathsf{tot}}-1]}$. □

Proof. We can prove hiding by relying on the hiding of the matrix commitment scheme and the zero-knowledge of the underlying CP-SNARKs[4].

By definition of extractable commitment we can interpret the commitment function as a first sub-procedure that generates a (binding) commitment and second procedure that generates a proof. By inspection of the algorithm, we can divide the commitment function above in this way. We prove knowledge soundness in the AGM for the derived CP-SNARK.

The extractor $\mathcal{E}_{\mathsf{DT}}$, using the algebraic representations, extract polynomials from the commitments $\mathsf{c}_{\_}, \mathsf{c}_{\_}, \mathsf{c}_v$, the commitments $\mathsf{c}'_{\_}, \mathsf{c}'_{\_}$, the sparse commitments $\mathsf{c}_L, \mathsf{c}_R$ and the commitment c_E. From such polynomials, the extractor can derive matrices $(\mathbf{N}^-, \mathbf{N}^-, \mathbf{L}, \mathbf{R}, \mathbf{E}, \mathbf{v})$. In particular, the matrix $\mathbf{N}^-$ is defined as the sum of the matrices $\bar{\mathbf{F}}_{\_}$ extracted from $\mathsf{c}_{\_}$ and $\bar{\mathbf{P}}_{\_}$ extracted from $\mathsf{c}'_{\_}$ (similarly for $\mathbf{N}^-$). We show that the constraints of theorem 6.6 hold, thus the extractor can compute a valid quasi-complete decision tree T from the extracted extended encoding. We proceed with a sequence of hybrids where $\mathbf{H}_0$ is the extractability game for CS_{DT} that returns 1 if the verifier accepts the proof and the extractor fails to produce a valid witness. In the next hybrids we sometimes reduce to the (N_1, N_2)-PDL assumption (see definition A.2).

Hybrid $\mathbf{H}_1$. The Hybrid $\mathbf{H}_1$ is the same as $\mathbf{H}_0$ but it additionally runs the extractor w.r.t. instances $([N_{\mathsf{int}}], \mathsf{c}_{\_})$, $([N_{\mathsf{int}}], \mathsf{c}_{\_})$, $((N_{\mathsf{int}}, N_{\mathsf{tot}}], \mathsf{c}'_{\_})$, $((N_{\mathsf{int}}, N_{\mathsf{tot}}], \mathsf{c}'_{\_})$ let $\tilde{\mathbf{F}}^-, \tilde{\mathbf{F}}^-, \tilde{\mathbf{P}}^-$ and $\tilde{\mathbf{P}}^-$ be the extracted matrices the hybrid returns 0 if either $\tilde{\mathbf{F}}^-$ or $\tilde{\mathbf{F}}^-$ have the first N_{int} rows set to $\mathbf{0}$ or $\tilde{\mathbf{P}}^-$ or $\tilde{\mathbf{P}}^-$ have the last $N_{\mathsf{tot}} - N_{\mathsf{int}}$ rows set to $\mathbf{0}$ or the matrices are not the valid opening for the respective commitments $\mathsf{c}_{\_}, \mathsf{c}_{\_}, \mathsf{c}'_{\_}$ and $\mathsf{c}'_{\_}$. It is easy to see that $\Pr[\mathbf{H}_0 = 1] \leq \Pr[\mathbf{H}_1 = 1] = \mathsf{negl}(n)$ where the negligible factor comes from the knowledge soundness of $\mathsf{CP}_{\mathsf{sm}}$.

Hybrid $\mathbf{H}_2$. The Hybrid $\mathbf{H}_2$ is the same as $\mathbf{H}_1$ but it additionally returns 0 if $\tilde{\mathbf{F}}^- \neq \mathbf{F}^-$ or $\tilde{\mathbf{F}}^- \neq \mathbf{F}^-$ or $\tilde{\mathbf{P}}^- \neq \mathbf{P}^-$ or $\tilde{\mathbf{P}}^- \neq \mathbf{P}^-$. Let E_1 be the event that one of previous dis-equations holds true. We have that $\Pr[\mathbf{H}_0 = 1] \leq \Pr[\mathbf{H}_1 = 1] + \Pr[E_1]$. Notice the event E_1 implies an attacker against the binding property of the matrix commitment scheme, thus $\Pr[E_1] = \mathsf{negl}(n)$.

In the next hybrids we can iteratively use the same proof strategy of the previous hybrids by first relying on the knowledge soundness of one of the CP-SNARKs and then on the binding property of the commitment scheme. Thus from now on we implicitly assume that the matrices extracted by the extractors of the CP-SNARKs match the matrices extracted by the extractor $\mathcal{E}_{\mathsf{DT}}$.

Hybrid $\mathbf{H}_3$. The Hybrid $\mathbf{H}_3$ is the same as $\mathbf{H}_2$ but it additionally returns 0 if $\mathbf{N}_1^- \neq \mathbf{0}$ or $\mathbf{N}_1^- \neq (B+1, \ldots, B+1)$. By the knowledge soundness of $\mathsf{CP}_{\mathsf{sm}}$ we have that both $\bar{\mathbf{F}}_{-1}$ and

[4] Notice that, using higher degrees for the randomizers of the commitments, hiding would still hold even if the proofs leaked evaluation points (see the notion of leaky-zero-knowledge from [5]) from the commitments in π

$\bar{\mathbf{P}}_{-1}$ equal the row vector $\mathbf{0}$, thus $\mathbf{N}^-_1 = \mathbf{0}$. Similarly for $\mathbf{N}^-_1$ however here we notice that we prove that $(\mathbf{N}^- + \mathbf{B})_1$ equals to $\mathbf{0}$ and $\bar{\mathbf{F}}_{-1}$ equals to $\mathbf{0}$ and thus $\mathbf{N}^-_1$ is the row vector $(B+1,\ldots,B+1)$.

Notice that when $\mathbf{H}_3 = 1$ the constraint eq. (6.17) of theorem 6.6 holds.

Hybrid $\mathbf{H}_4$. The Hybrid $\mathbf{H}_4$ is the same as $\mathbf{H}_3$, but it additionally (runs the extractor w.r.t. instance-proof tuple $(\mathsf{c}_L, \mathsf{c}_{N,-}, \mathsf{c}'_-), \pi_1$, such extractor exists because of the knowledge soundness of $\mathsf{CP}_{\mathsf{lin}}$) and returns 0 if $\mathbf{L}\cdot\mathbf{N}^- \neq \mathbf{P}^-$. The distinghuishing event between $\mathbf{H}_3$ and $\mathbf{H}_4$ is the event the extractor fails to extract a valid witness, thus if $\mathsf{CP}_{\mathsf{lin}}$ is knowledge-sound (and the matrix commitment is binding) then $\Pr[\mathbf{H}_3] \leq \Pr[\mathbf{H}_4] + \mathsf{negl}(n)$.

Hybrid $\mathbf{H}_5$. The hybrid $\mathbf{H}_5$ is the same as $\mathbf{H}_4$, but it additionally (runs the extractor w.r.t. instance-proof tuple $(\mathsf{c}_R, \mathsf{c}_{N,-}, \mathsf{c}'_-), \pi_3$) returns 0 if $\mathbf{R}\cdot\mathbf{N}^- \neq \mathbf{P}^-$. Similarly to the previous hybrids, this hybrid is negligibly close to $\mathbf{H}_4$ because of the knowledge-soundness of $\mathsf{CP}_{\mathsf{lin}}$ (and the binding property of the commitment scheme).

Notice that when $\mathbf{H}_5 = 1$ the constraint eq. (6.18) of theorem 6.6 holds.

Hybrid $\mathbf{H}_6$. The hybrid $\mathbf{H}_6$ is the same as $\mathbf{H}_5$, but it additionally returns 0 if $\bar{\mathbf{L}}\cdot\mathbf{N}^-$ is not a valid opening for c_{ln} or $\bar{\mathbf{R}}\cdot\mathbf{N}^-$ is not a valid opening for c_{rn}. We can prove $\Pr[\mathbf{H}_5 = 1] \leq \Pr[\mathbf{H}_6 = 1] + \mathsf{negl}(n)$ based on the knowledge soundness of $\mathsf{CP}_{\mathsf{lin}}$.

Hybrid $\mathbf{H}_7$. The hybrid $\mathbf{H}_7$ is the same as $\mathbf{H}_6$, but it additionally returns 0 if

$$\bar{\mathbf{E}}\cdot(\bar{\mathbf{L}}\cdot\mathbf{N}^- - \bar{\mathbf{R}}\cdot\mathbf{N}^-) \neq \mathbf{0} \vee \mathbf{1} - \bar{\mathbf{E}}\cdot(\mathbf{P}^- - \bar{\mathbf{R}}\cdot\mathbf{N}^-) \neq \mathbf{0} \vee \mathbf{1} - \bar{\mathbf{E}}\cdot(\mathbf{P}^- - \bar{\mathbf{L}}\cdot\mathbf{N}^-) \neq \mathbf{0}.$$

Leveraging the conditions from $\mathbf{H}_6$ on c_{ln} and c_{rn}, using knowledge soundness of $\mathsf{CP}_{\mathsf{had}}$ and the binding of the matrix commitment scheme we have $\Pr[\mathbf{H}_6 = 1] \leq \Pr[\mathbf{H}_7 = 1] + \mathsf{negl}(n)$.

Notice that when $\mathbf{H}_7 = 1$ the constraints of eqs. (6.19) and (6.20) of theorem 6.6 hold.

Hybrid $\mathbf{H}_8$. The Hybrid $\mathbf{H}_8$ is the same as $\mathbf{H}_7$, but it additionally returns 0 if $\mathbf{N}^- - \mathbf{N}^- - \mathbf{1} \notin [B]^{N_{\mathsf{tot}}\times d}$. By the binding property of the commitment scheme and by the knowledge soundness of $\mathsf{CP}_{\mathsf{range}}$ we have $\Pr[\mathbf{H}_7] \leq \Pr[\mathbf{H}_8] + \mathsf{negl}(n)$.
Notice that when $\mathbf{H}_8 = 1$ we have that $\mathbf{N}^-_{i,j} < \mathbf{N}^-_{i,j}$ for any i, j (namely, the constraint in item b) of theorem 6.6 holds).

Hybrid $\mathbf{H}_9$. The Hybrid $\mathbf{H}_9$ is the same as $\mathbf{H}_8$, but it additionally extracts the (sparse) matrices $\bar{\mathbf{L}}, \underline{\mathbf{R}}$ and returns 0 if $\bar{\mathbf{L}} + \underline{\mathbf{R}}$ is not a permutation of the matrix $(\mathbf{0} \| \mathbf{I}_{N_{\mathsf{tot}}-1})$. We reduce to knowledge soundness of $\mathsf{CP}_{\mathsf{perm}}$ noticing that the polynomial $id(X)$ in the indexer of the instance $((N_{\mathsf{tot}}-1, id), \mathsf{c}_L + \mathsf{c}'_R)$ is a valid representation (according to the sparse matrix commitment scheme) of the sparse-matrix $(\mathbf{0} \| \mathbf{I}_{N_{\mathsf{tot}}-1})$.

Hybrid $\mathbf{H}_{10}$. The Hybrid $\mathbf{H}_{10}$ is the same as $\mathbf{H}_9$, but it returns 0 if $\underline{\mathbf{R}}$ is not a shift of $\bar{\mathbf{R}}$. By the knowledge soundness of $\mathsf{CP}_{\mathsf{shift}}$ we have that the probability of the two hybrids is negligibly close.

Notice that if $\mathbf{H}_{10} = 1$ then all constraints in theorem 6.6 are satisfied thus there must exist a valid quasi-complete decision tree T with N_{tot} nodes associated to the extracted matrices returned by the extractor, which is in contradiction with the winning condition of the adversary, thus the probability of $\mathbf{H}_{10} = 1$ is equal to 0. □

A.5.3. CP-SNARK FOR STATISTICS ON DECISION TREES

Theorem 6.8. $\mathsf{CP}_{DT} = (\mathsf{Derive}, \mathsf{Prove}, \mathsf{Verify})$ *in fig. 6.4 defines an Universal CP-SNARK for the indexed CP-relation* $\hat{\mathcal{R}}_{\mathsf{DTstat}}$.

Proof. Zero-knowledge follows easily from the hiding of the commitments $\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3$ and the zero-knowledge of the three CP-SNARKs. In particular, the zero-knowledge simulator could sample the commitments by commiting to dummy values and run the zero-knowledge simulators of the three CP-SNARKs.

We recall that $\mathsf{c}_{\_}$ (resp. $\mathsf{c}_{\_}$) commits to the matrix $\bar{\mathbf{F}}_{\_}$ (resp. $\bar{\mathbf{F}}_{\_}$) whose first N_{int} rows are filled with 0 and the remaining submatrix is $\mathbf{F}^{\_}$ (resp. $\mathbf{F}^{\_}$).

The completeness follows by theorem 6.5, the homomorphic properties of the matrix commitment scheme and the completeness of the CP-SNARKs. In particular, the lemma implies that c_3 commits to the vector $(\mathsf{T}(\mathbf{x}_j))_{j\in[m]}$, moreover by definition of k_{T}, the matrix $(\mathbf{X} - \bar{\mathbf{F}}_{\_|K})$ contains non negative numbers smaller than B and the matrix $(\bar{\mathbf{F}}_{\_|K} - \mathbf{X} - \mathbf{1})$ contains non negative numbers smaller than B.

For knowledge soundness, we define the extractor of the CP-SNARK to be the same as the extractor $\mathcal{E}_{\mathsf{Com}}$ of the extractable commitment scheme. We proceed with a sequence of hybrids where $\mathbf{H}_0$ is the knowledge soundness game for the CP_{DT} with extractor $\mathcal{E}_{\mathsf{Com}}$.

Hybrid $\mathbf{H}_1$. Let $\tilde{\mathsf{T}}$ (and opening material $(\rho_v, \rho_{\_}, \rho_{\_})$) be the extracted quasi-complete decision tree, the hybrid $\mathbf{H}_1$ additionally computes $(\mathbf{N}^{\_}, \mathbf{N}^{\_}, \mathbf{v}, \mathbf{L}, \mathbf{R}, \mathbf{E}) \leftarrow \mathsf{Encode}(\tilde{\mathsf{T}})$ and sets $\tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{F}}_{\_}$ be the sub-matrices (relative to the leaf) of $\mathbf{N}^{\_}, \mathbf{N}^{\_}$ and returns 0 if $\mathbf{v}, \tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{F}}_{\_}$ (and their opening materials) do not commit to $\mathsf{c}_v, \mathsf{c}_{\_}, \mathsf{c}_{\_}$. It is easy to see that $\Pr[\mathbf{H}_0] \leq \Pr[\mathbf{H}_1] + \mathsf{negl}(n)$, where the latter negligible value depends on the error of the extractable decision-tree commitment scheme.

Hybrid $\mathbf{H}_2$. The hybrid $\mathbf{H}_2$ is the same as $\mathbf{H}_1$ but it additionally runs the extractor of $\mathsf{CP}_{\mathsf{lookup}*}$ extracting matrices $\tilde{\mathbf{M}}_1, \tilde{\mathbf{M}}_2, \tilde{\mathbf{m}}_3, \tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{v}}$ and it outputs 0 if $(\tilde{\mathbf{M}}_1 \| \tilde{\mathbf{M}}_2 \| \tilde{\mathbf{m}}_3) \not\prec (\tilde{\mathbf{F}}_{\_} \| \tilde{\mathbf{F}}_{\_} \| \tilde{\mathbf{v}})$. It is easy to see that $\Pr[\mathbf{H}_1] \leq \Pr[\mathbf{H}_2] + \mathsf{negl}(n)$, where the latter negligible value depends on the knowledge soundness error of $\mathsf{CP}_{\mathsf{lookup}*}$.

Hybrid $\mathbf{H}_3$. The hybrid $\mathbf{H}_3$ is the same as $\mathbf{H}_2$ but it additionally returns 0 if $(\tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{F}}_{\_}, \tilde{\mathbf{v}}) \neq (\bar{\mathbf{F}}_{\_}, \bar{\mathbf{F}}_{\_}, \mathbf{v})$, where the former matrices are extracted from the extractor of $\mathsf{CP}_{\mathsf{lookup}*}$ and the latter from the extractor of the extractable commitment. It is easy to see that $\Pr[\mathbf{H}_2] \leq \Pr[\mathbf{H}_3] + \mathsf{negl}(n)$, because the distinguishing event allows to break the binding property of the matrix commitment scheme.

Hybrid $\mathbf{H}_4$. The hybrid $\mathbf{H}_4$ is the same as $\mathbf{H}_3$ but that additionally returns 0 if $(\mathbf{X} - \tilde{\mathbf{M}}_1) \notin [B]^{m\times d}$. To show $\Pr[\mathbf{H}_3] \leq \Pr[\mathbf{H}_4] + \mathsf{negl}(n)$ we can follow the same two-fold strategy of the previous two hybrids, namely we can (1) define a sub-hybrid experiment where we run the extractor of $\mathsf{CP}_{\mathsf{range}}$ and return 0 if the extracted matrix is not in the range thus reducing to the extractability of $\mathsf{CP}_{\mathsf{range}}$ and (2) we can show that the extracted matrix

must be equal to $\mathbf{X}-\tilde{\mathbf{M}}_1$ because of the binding and homomorphic property of the matrix commitment scheme.

Hybrid $\mathbf{H}_5$. Similarly to the previous item, the hybrid $\mathbf{H}_5$ additionally returns 0 if $(\tilde{\mathbf{M}}_2 - \mathbf{X} - \mathbf{1}) \notin [B]^{m\times d}$. We can show $\Pr[\mathbf{H}_4] \leq \Pr[\mathbf{H}_5] + \mathsf{negl}(n)$ in a very similar manner to the previous step.

Hybrid $\mathbf{H}_6$. Let K be the set of indexes such that $(\tilde{\mathbf{M}}_1 \| \tilde{\mathbf{M}}_2 \| \tilde{\mathbf{m}}_3) = (\tilde{\mathbf{F}}_{-} \| \tilde{\mathbf{F}}_{-} \| \mathbf{v})_{|K}$. The hybrid $\mathbf{H}_6$ additionally returns 0 if $K \neq \{k_{\bar{\mathsf{T}}}(\mathbf{x}_1), \ldots, k_{\bar{\mathsf{T}}}(\mathbf{x}_m)\}$. By the change introduced in $\mathbf{H}_1$ and by theorem 6.5 for any $i \neq j$ the boxes $(\mathbf{F}^{-}{}_i, \mathbf{F}^{-}{}_i)$ and $(\mathbf{F}^{-}{}_j, \mathbf{F}^{-}{}_j)$ do not overlap. Thus for any $i \in [m]$ there must exists only one index k_i such that $(\tilde{\mathbf{M}}_1)_i = \tilde{\mathbf{F}}_{-k_i}$ (resp. $(\tilde{\mathbf{M}}_2)_i = \tilde{\mathbf{F}}_{-k_i}$), moreover by the changes introduced in $\mathbf{H}_4$ and $\mathbf{H}_5$ we have that $\tilde{\mathbf{F}}_{-k_i} \leq \mathbf{x}_i < \tilde{\mathbf{F}}_{-k_i}$, thus such a unique index k_i must be equal to $k_{\bar{\mathsf{T}}}(\mathbf{x}_i)$. We have $\Pr[\mathbf{H}_6] = \Pr[\mathbf{H}_5]$.

Hybrid $\mathbf{H}_7$. The hybrid $\mathbf{H}_7$ additionally returns 0 if $y \neq S((\tilde{\mathbf{m}}_3)_1, \ldots, (\tilde{\mathbf{m}}_3)_m)$. Similarly previous hybrids we can reduce to the biding of the vector commitment to prove that the vector $\tilde{\mathbf{m}}_3$ is the same that the knowledge extractor of $\mathsf{CP}_{\mathsf{stat}}$ would compute and then reduce to the knowledge soundness of $\mathsf{CP}_{\mathsf{stat}}$. Thus, we have $\Pr[\mathbf{H}_6] \leq \Pr[\mathbf{H}_7] + \mathsf{negl}(n)$.

We show that the probability for $\mathbf{H}_7$ is 0, we can now conclude the proof by chaining the equations proved in the previous steps. Because of the changes in $\mathbf{H}_2$ and $\mathbf{H}_3$, in $\mathbf{H}_7$ we have that $\tilde{\mathbf{m}}_3 = \mathbf{v}_{|K}$, moreover, by the check introduced in $\mathbf{H}_6$ we have that $K = \{k_{\bar{\mathsf{T}}}(\mathbf{x}_1), \ldots, k_{\bar{\mathsf{T}}}(\mathbf{x}_m)\}$. Notice this already implies that $\forall j : \mathsf{T}(\mathbf{x}_j) \neq \perp$. Moreover, because of the check introduced in $\mathbf{H}_7$ we have $y = S(v_{k_{\bar{\mathsf{T}}}(\mathbf{x}_1)}, \ldots, v_{k_{\bar{\mathsf{T}}}(\mathbf{x}_m)})$. These last two implications negate the winning condition of the knowledge soundness of CP_{DT}, thus $\Pr[\mathbf{H}_7] = 0$. □

A.5.4. CP-SNARKs FOR LINEAR RELATIONS WITH SPARSE MATRIX COMMITMENT

Let $\mathbf{M}$ be a *basic matrix*, namely a matrix whose rows are elementary vectors. Let $\mathbb{H}$ be the subgroup of $\mathbb{F}$ generated by h and defined in the commitment key for the vector commitment[5]. For any basic matrix $\mathbf{M} \in \{0,1\}^{k\times n}$ and $k, n \in \mathbb{N}$, let $col_{\mathbf{M}}(X)$ be the (low-degree) polynomial such that $col_{\mathbf{M}}(h^i) = h^j$ where the i-th row of $\mathbf{M}$ is the vector $\mathbf{e}_j^\top$ (notice that $col_{\mathbf{M}}$ can also be interpreted as a vector whose i-th element is the value h^j). We define the sparse (hiding) commitment of a matrix $\mathbf{M}$ as a (hiding) polynomial commitment of $col_{\mathbf{M}}$. Namely, we define:

$$\mathsf{sparseCom}(\mathsf{ck}, \mathbf{M}, \rho) := \mathsf{Com}(\mathsf{ck}, col_{\mathbf{M}}, \rho).$$

We write $\boxed{\mathbf{M}}$ to underline that the matrix $\mathbf{M}$ is committed with a sparse matrix commitment. Our goal is to realize a CP-SNARK for the relation:

$$\hat{\mathcal{R}}_{\mathsf{lin}} = \left\{ \mathsf{pp}; \varepsilon; (\boxed{M}, \mathbf{N}, \mathbf{R}) : \mathbf{M}\cdot\mathbf{N} = \mathbf{R}, \ \ \mathbf{N} \in \mathbb{F}^{n\times d} \right\}.$$

[5] We assume $|\mathbb{H}| \geq k, n$.

Our building blocks are CP-SNARKs CP' and CP'' for the relations:

$$\hat{\mathcal{R}}'_{\mathsf{lin}} = \left\{ \mathsf{pp}; \varepsilon; (\boxed{M}, \mathbf{n}, \mathbf{r}) : \mathbf{M} \cdot \mathbf{n} = \mathbf{r}, \ \ \mathbf{n} \in \mathbb{F}^{n \cdot d} \right\},$$
$$\hat{\mathcal{R}}''_{\mathsf{lin}} = \left\{ \mathsf{pp}; \mathbf{M}; \varepsilon; (\mathbf{n}, \mathbf{r}) : \mathbf{M} \cdot \mathbf{n} = \mathbf{r}, \ \ \mathbf{n} \in \mathbb{F}^{n \cdot d} \right\}.$$

Notice that above $\mathbf{n}$ is a vector, while n is an integer. The difference between the two relations is that in the first $\mathbf{M}$ is part of the witness while in the second $\mathbf{M}$ is part of the index. Notice that an instantiation of CP' can be found in Baloo [6] while instantiations of CP'' can be derived easily from zkSNARKs for R1CS based on holographic polynomial IOP such as [5, 7, 8]. The prover time complexity for the latter scheme depends quasi-linearly on the sparsity of the matrix $\mathbf{M}$.

Gadget Matrices, Operators and Vectorizations. Consider the matrix $\bar{\mathbf{I}}_{n,d}$ which stacks d identity matrices $\mathbf{I}_n$ of size n, namely $\bar{\mathbf{I}}_{n,d} = \mathbf{I}_n \otimes \mathbf{1}$ where $\mathbf{1}$ is of length d. Consider the linear operator r that maps a matrix $\mathbf{A}$ to the vectorization row-by-row of $\mathbf{A}$, similarly, consider the linear operator c that maps a matrix $\mathbf{A}$ to the vectorization column-by-column of $\mathbf{A}$. Finally, we consider the permutation matrix $\mathbf{P}$ such that for any $\mathbf{A}$ we have:

$$\mathbf{P} \cdot \mathsf{r}(\mathbf{A}) = \mathsf{c}(\mathbf{A}) \tag{A.4}$$

We also recall that to compute a matrix commitment of $\mathbf{A}$ we are implicitly computing a vector commit to $\mathsf{r}(\mathbf{A})$.

Let $\mathbf{M} \otimes \mathbf{I}$ be the tensor-product of $\mathbf{M}$ and $\mathbf{I}$, namely the following matrix:

$$\mathbf{M} \otimes \mathbf{I} = \begin{bmatrix} \mathbf{M} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{M} \end{bmatrix}$$

It is not hard to prove that the following holds:

$$\mathbf{M} \cdot \mathbf{N} = \mathbf{R} \iff (\mathbf{M} \otimes \mathbf{I}) \cdot \mathsf{c}(\mathbf{N}) = \mathsf{c}(\mathbf{R}). \tag{A.5}$$

Moreover if $\mathbf{M} \in \{0,1\}^{k \times n}$ is a basic matrix then :

$$col_{\mathbf{M} \otimes \mathbf{I}}(h^{k \cdot i + j}) = col_{\mathbf{M}}(h^j) + n \cdot i.$$

Namely, the $(k \cdot i + j)$-row of $\mathbf{M} \otimes \mathbf{I}$ contains the j-th row vector of $\mathbf{M}$ *shifted* of $n \cdot i$ columns. The equation above can be translated in the vector domain. Namely, if we let $\mathbf{c}$ the evaluation over $\mathbb{H}$ of $col_{\mathbf{M}}(X)$ and $\mathbf{c}'$ the evaluation over $\mathbb{H}$ of $col_{\mathbf{M} \otimes \mathbf{I}}(X)$ we have that:

$$\mathbf{I}_{n,d} \cdot \mathbf{c} = \mathbf{c}' - \mathbf{s}$$

where the *shifting vector* $\mathbf{s}$ is such that $s_{k \cdot i + j} = n \cdot i$ for all $i, j \in \mathbb{N}$ and $j < k$. Thus for any basic matrix $\mathbf{M} \in \{0,1\}^{k \times n}$ and for any d we have:

$$\mathbf{M}' = \mathbf{M} \otimes \mathbf{I}_d \iff \bar{\mathbf{I}}_{n,d} \cdot col_{\mathbf{M}} = col_{\mathbf{M}'} - \mathbf{s}. \tag{A.6}$$

Our Scheme. Our CP-SNARK scheme is shown in fig. A.7.

$\mathsf{KGen}(\mathsf{ck})$**:**

Run the keygen algorithms of CP' and CP''. Moreover, derive proving and verification keys for the matrix $\bar{\mathbf{I}}_{n,d}$ and for the matrix $\mathbf{P}$.

$\mathsf{Prove}(\mathsf{srs}, (\mathsf{c}_M, \mathsf{c}_N, \mathsf{c}_R), (M, \mathbf{N}, \mathbf{R}), (\rho_M, \rho_N, \rho_R))$**:**

Commit $\mathsf{c}_{R,c} \leftarrow \mathsf{Com}(\mathsf{ck}, \mathsf{c}(\mathbf{R}))$, $\mathsf{c}_{N,c} \leftarrow \mathsf{Com}(\mathsf{ck}, \mathsf{c}(\mathbf{N}))$.
Prove that $(\mathbf{P}; \varepsilon; \mathsf{r}(\mathbf{R}), \mathsf{c}(\mathbf{R})) \in \hat{\mathcal{R}}''_{\mathsf{lin}}$.
Prove that $(\mathbf{P}; \varepsilon; \mathsf{r}(\mathbf{N}), \mathsf{c}(\mathbf{N})) \in \hat{\mathcal{R}}''_{\mathsf{lin}}$.
Compute $\mathbf{M}' = \mathbf{M} \otimes \mathbf{I}$ and commit $\mathsf{c}_{M'} \leftarrow \mathsf{sparseCom}(\mathsf{ck}, \mathbf{M}')$.
Prove $(\bar{\mathbf{I}}_{n,d}; \varepsilon; col_{\mathbf{M}}, col_{\mathbf{M}'} - \mathbf{s}) \in \hat{\mathcal{R}}''_{\mathsf{lin}}$.
Prove $(\varepsilon; M', \mathsf{c}(\mathbf{N}), \mathsf{c}(\mathbf{R})) \in \mathcal{R}'_{\mathsf{lin}}$.

$\mathsf{Verify}(\mathsf{srs}, (\mathsf{c}_M, \mathsf{c}_N, \mathsf{c}_R), \pi)$**:**

Parse $\pi = (\mathsf{c}_{M'}, \mathsf{c}_{R,c}, \pi_1, \pi_2, \pi_3)$.
Verify the proofs:

1. π_1 with instance $(\mathsf{c}_R, \mathsf{c}_{R,c})$ and verification key for $\mathbf{P}$ (for CP')
2. π_2 with instance $(\mathsf{c}_N, \mathsf{c}_{N,c})$ and verification key for $\mathbf{P}$ (for CP')
3. π_3 with instance $(\mathsf{c}_{M'} - \mathsf{Com}(\mathsf{ck}, \mathbf{s}), \mathsf{c}_M)$ and verification key for $\bar{\mathbf{I}}_{n,d}$ (for CP').
4. π_4 with instance $(\mathsf{c}_{M'}, \mathsf{c}_{N,c}, \mathsf{c}_{R,c})$ (for CP'').

Figure A.7: Our CP-SNARK scheme for linear relations with sparse matrix commitments.

Theorem A.8. *The CP-SNARK defined in fig. A.7 is zero-knowledge and knowledge sound.*

Proof Sketch. Zero-knowledge is trivially implied by the zero-knowledge of the CP-SNARK CP' and CP'' and by the hiding property. As for knowledge soundness, we let the extractor be the same of CP'' on instance $(\mathsf{c}_{M'}, \mathsf{c}_{N,c}, \mathsf{c}_{R,c})$. Because of the knowledge soundness of π_3 and by eq. (A.6) we have that the extracted matrix $\mathbf{M}'$ is of the form $\mathbf{M}' = \mathbf{M} \otimes \mathbf{I}$. Because of the knowledge soundness of π_1 and π_2 and eq. (A.4) we have that the commitments $\mathsf{c}_{N,c}$ and $\mathsf{c}_{R,c}$ commits to $\mathsf{c}(\mathbf{N})$ and $\mathsf{c}(\mathbf{R})$. Finally, because of the knowledge soundness of π_4 we have that $\mathbf{M}' \cdot \mathsf{c}(\mathbf{N}) = \mathsf{c}(\mathbf{R})$ and thus because of eq. (A.5) we have that the extracted witness (once parsed adequately) is valid for the relation. □

A.6. EFFICIENCY BREAKDOWN FOR OUR MATRIX LOOKUP ARGUMENTS

- **Proof size.** Our protocol requires one zkcq^+ proof $(9\mathfrak{g}_1 + 1\mathfrak{f})$, plus: three KZG commitments, four group elements (π_R and $\pi_{R'}$), and one field element. The total proof size is $16\mathfrak{g}_1 + 2\mathfrak{f}$.

- **Proving time.** In addition to the zkcq^+ prover (requiring $O(nd)$ group operations and $O(nd\log(nd))$ field operations) the prover performs:

  – field operations required to compute polynomial evaluations for w, σ and $v_{\mathbb{K}}(X)$. We have $\deg(w) = n$, $\deg(\sigma) = n \cdot d$ and $\deg(v_{\mathbb{K}}(X)) = n \cdot d$.

  - three multiexponentiations of size n in order to compute $[\sigma(s)]_1, [\sigma(\kappa \cdot s)]_1$, $[w(s)]_1$; notice that we use the sparsity of σ (see definition of σ) (we ignore the masking factors for simplicity).
  - four multiexponentiations of size $n \cdot d$ in order to compute group elements π_R and $\pi_{R'}$ (two batched KZG proofs in zero-knowledge).

- **Verification time.** In addition to the steps for zkcq$^+$'s verifier (requiring 7 pairings), verification requires: a constant number of group operations and six pairings (we use the fact that we can batch some of the pairing equations). The total number of pairings performed by the verifier is 13.

A.7. DETAILS ON THE EXPERIMENTAL EVALUATION

Here we provide details on the machines, their running times, and the concrete efficiency analysis for estimates.

For the experimental parameters referred to in the main text:

- Our proving time if run on our machine (hereafter OM) is below $4s$.
- Our proving time if run on machine from [3] (hereafter TM) is below $13s$ (at least 9x faster than the approach in [3])

We stress that these are quite pessimistic upper bounds. Below we detail how these numbers are derived.

A.7.1. DETAILS ON THE MACHINES AND THEIR RUNNING TIMES

A setup of the machines can be found at:

https://aws.amazon.com/ec2/instance-types/

As mentioned in footnote [13],OM is an EC2 c5.9xlarge and TM is an EC2 c5n.2xlarge.

On OM, an MSM of size 100K in $\mathbb{G}_1$ is approximately 140ms (using arkworks; we derived these numbers running the Zkalc framework on OM[6]). On TM, the same MSM requires approx 450ms.

A.7.2. DETAILS ON ANALYSIS AND ESTIMATES

We can break down the proving time in the following sub-pieces:

Total time = time(commitments) + time(zklookup) + time(zkrange) + time(zkstat)

Parameters of interest: $d = 50$, $m = 2000$ and $k = 10$ (see paragraphs starting on page 164 for definitions). Below MSM(x) stands for one multi-scalar multiplication of size x. To estimate field operations we use the conservative cost assumption $4\mathbb{F}\text{ops} \leq 1\mathbb{G}_1\text{ops}$. MSMs are in $\mathbb{G}_1$.

TIME(COMMITMENTS):

- $c_1, c_2, c_3, c_{\mathbf{X}}$: 2MSM(md) + 1MSM(d) + 1MSM(md)

[6] https://zka.lc/

TIME(ZKLOOKUP):

- $zkcq+$: 7MSM(md) (see section 6.4.1, section 6.4.2 and appendix A.3)
- matrix lookup (non-zkcq+ part of the proof): 3MSM(m) + 4MSM(md) (see fig. 6.2 and appendix A.6)

TIME(ZKRANGE):

- Batched cq+: 7MSM(md) (see section 6.4.1)

TIME(ZKSTAT):

- The main computation in zkstat involves proving relation in Equation 6.27. Even with an R1CS-based (Rank-1 Constraint System) general purpose SNARK, this relation can be implemented very efficiently For example, an upper bound on a naive implementation is $\approx m \cdot k$ (with a very small multiplicative constant). This number accounts for implementing the equality predicate (bit decomposition and bit equality checks, done m times) and a sum of m bits (which can be described with a single constraint row in an R1CS). For representative values of m and k—respectively 2000 and 10 (see Figure 5 in [3])—this roughly corresponds to 20K constraints which results in an additional proving time of only tens of milliseconds. .

FINAL SUM:

Summing the above, we obtain: 17MSM(md) + 3MSM(m) + 1MSM(d). This quantity is for *cryptographic* operations. In order to also account for field operations, we apply our cost assumption from above and add an additional 25% cost. The concrete numbers we obtain are those stated above.

REFERENCES

[1] J. Groth. "On the Size of Pairing-Based Non-interactive Arguments". In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 305–326.

[2] A. Chiesa, Y. Hu, M. Maller, P. Mishra, P. Vesely, and N. P. Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*. Vol. 12105. Lecture Notes in Computer Science. Springer, 2020, pp. 738–768.

[3] J. Zhang, Z. Fang, Y. Zhang, and D. Song. "Zero Knowledge Proofs for Decision Tree Predictions and Accuracy". In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. ACM, 2020, pp. 2039–2053.

[4] H. Lipmaa. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*. Vol. 7194. Lecture Notes in Computer Science. Springer, 2012, pp. 169–189.

[5] M. Campanelli, A. Faonio, D. Fiore, A. Querol, and H. Rodríguez. "Lunar: A Toolbox for More Efficient Universal and Updatable zkSNARKs and Commit-and-Prove Extensions". In: *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III*. Vol. 13092. Lecture Notes in Computer Science. Springer, 2021, pp. 3–33.

[6] A. Zapico, A. Gabizon, D. Khovratovich, M. Maller, and C. Ràfols. "Baloo: Nearly Optimal Lookup Arguments". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1565.

[7] J. Groth, M. Kohlweiss, M. Maller, S. Meiklejohn, and I. Miers. "Updatable and Universal Common Reference Strings with Applications to zk-SNARKs". In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*. Vol. 10993. Lecture Notes in Computer Science. Springer, 2018, pp. 698–728.

[8] C. Ràfols and A. Zapico. "An Algebraic Framework for Universal and Updatable SNARKs". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, pp. 774–804.

CURRICULUM VITÆ

Tianyu LI

Tianyu Li was born in Jinan, Shandong Province, China, on 17 January 1996. After spending an early age in Jinan, he moved to Shanghai and graduated from Shanghai High School in 2014. Then, he obtained his bachelor's degree in Information Security from Shanghai Jiao Tong University, China, in 2018. After that, he received his master's degree in Informatics from the University of Edinburgh, United Kingdom, in 2019.

Since 2020, Tianyu Li has started as a PhD student at Delft University of Technology under the supervision of Prof.dr.ir. R.L. Lagendijk and Dr. Z. Erkin. His PhD position is funded by the Dutch Research Council (NWO) under the *Spark! Living Lab project*. In 2022, he also visited EURECOM in France as a guest researcher for three months under the supervision of Prof.dr. Melek Önen and Dr. Antonio Faonio. His main research is on data privacy protection in supply chains and machine learning using differential privacy and applied cryptography. During his PhD, he supervised four master students and five bachelor students. He also contributed to the teaching assistance work of three master courses. In academia, he served as a PC member for ICASSP, AAMAS and IWSEC.

List of Publications

Journal

2. R. Kromes, **T. Li**, M. Bouillon, T. E. Güler, V. Hulst, Z. Erkin. "Fear of Missing Out: Constrained Trial of Blockchain in Supply Chain", in *Sensors* 24(3), pp. 986-1010, 2024.

1. **T. Li**, Z. Erkin, R.L. Lagendijk. "Privacy-Preserving Bin-Packing with Differential Privacy", in *IEEE Open Journal of Signal Processing* vol. 3, pp. 94-106, 2022.

Conference and Workshop

5. M. Campanelli, A. Faonio*, D. Fiore, **T. Li***, H. Lipmaa (in alphabetical order), "Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees", in *Public-Key Cryptography – PKC 2024*, pp. 337-369, 2024.

4. **T. Li**, L. Xu, Z. Erkin, R.L. Lagendijk. "Trajectory Hiding and Sharing for Supply Chains with Differential Privacy", in *28th European Symposium on Research in Computer Security (ESORICS 2023)*, pp. 297-317, 2023.

3. L. Xu, **T. Li**, Z. Erkin. "Verifiable Credentials with a Privacy-Preserving Tamper-Evident Revocation Mechanism", in *Fifth International Conference on Blockchain Computing and Applications (BCCA 2023)*, pp. 266-273, 2023.

2. **T. Li**, J. Vos, Z. Erkin. "Decentralized Private Freight Declaration & Tracking with Data Validation", in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom 2022 Workshops)*, pp. 267-272, 2022.

1. D. Kester, **T. Li**, Z. Erkin. "PRIDE: A Privacy-Preserving Decentralised Key Management System", in *2022 IEEE International Workshop on Information Forensics and Security (WIFS 2022)*, pp. 1-6, 2022.

Preprint and Under Review

3. **T. Li**, R. Kromes, Z. Erkin. "PrivTrack: Privacy-Preserving Trajectory Tracking for Supply Chains", *under review.*

2. **T. Li**, D. van Tetering, Z. Erkin. "Robust Small-Scale Collaborative Learning against Inference Attacks Using Multi-Party Selection and Differential Privacy", *under review.*

1. D. Vos, J. Vos, **T. Li**, Z. Erkin, S. Verwer. “Differentially-Private Decision Trees with Probabilistic Robustness to Data Poisoning", *arXiv preprint arXiv:2305.15394*, 2023.