Trajectory Hiding and Sharing for Supply Chains with Differential Privacy

Li, Tianyu; Xu, Li; Erkin, Zekeriya; Lagendijk, Reginald L.

DOI
10.1007/978-3-031-51476-0_15

Publication date
2024

Document Version
Final published version

Published in
Computer Security – ESORICS 2023 - 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25–29, 2023, Proceedings

Trajectory Hiding and Sharing for Supply Chains with Differential Privacy

Tianyu Li(✉), Li Xu, Zekeriya Erkin, and Reginald L. Lagendijk

Cyber Security Group, Delft University of Technology, Delft, The Netherlands
{tianyu.li,l.xu-11,z.erkin,r.l.lagendijk}@tudelft.nl

Abstract. 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.

Keywords: Privacy-preserving · Differential privacy · Location privacy · Applied cryptography · Blockchain

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 [4]. Meanwhile, logistic companies provide a time slot for delivery. Unfortunately, these time slots usually span multiple hours, which reduces customer satisfaction on many levels [1]. 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

G. Tsudik et al. (Eds.): ESORICS 2023, LNCS 14345, pp. 297–317, 2024.
https://doi.org/10.1007/978-3-031-51476-0_15

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 [13]. Besides, the system is useful for disputes and knowing the reason for delays, and unburdening the customer service department [13]. It is already possible to see the use of real-time tracking, e.g., *DHL* offers a live tracking service for selected shipments [6]. Unfortunately, sharing accurate locations introduces security and privacy issues. According to [11,27], 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 [11,27].

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 [15], adding dummy data [16], applying suppression [32], or using differential privacy [33] 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 [7,8] 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 [2,33,35,36], 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 [5]. 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 [34]. Blockchain is a candidate for the decentralized solution since it is traceable, immutable and transparent [25].

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 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.

2 Preliminaries

Differential Privacy. Differential privacy (DP) was raised by Dwork [7,8] to protect individual privacy and better use the dataset. In Eq. 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 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 $O \subseteq range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O] + \delta \, . \tag{1}$$

The Gaussian mechanism is a widely used mechanism to achieve (ϵ, δ)-differential privacy [10], 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. [2] 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 2 (geo-indistinguishability). *An algorithm $\mathcal{A}$ satisfies ϵ-geo-indistinguishability iff for any two different points x, x':*

$$d_{\mathcal{P}}(\mathcal{A}(x), \mathcal{A}(x')) \leq \epsilon \cdot d(x, x') \, . \tag{2}$$

$d(\cdot, \cdot)$ 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. [2] 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:

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \leq \epsilon \, , \tag{3}$$

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 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.

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. [9] 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. [21] 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. [12] 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. [35] 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 [19]. Xiong et al. [36] applied differential privacy to cluster and select location points, but the whole trajectory is known before the perturbation. Andrés et al. [2] 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 [25], with which the participants can trace the transaction. Maouchi et al. [22] 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. [29] 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.

5 Location Perturbation

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

Algorithm 1: 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 6.
2: Get ϵ' using Equation 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.

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 (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:

$$l(x, f_i) = \begin{cases} l_s, & \text{if } d(x, f_i) \text{ is large} \\ l_m, & \text{if } d(x, f_i) \text{ is medium} \\ l_l, & \text{if } d(x, f_i) \text{ is small} , \end{cases} \tag{4}$$

where $d(x, f_i)$ is the distance between the location of truck x and receiver f_i. 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:

$$r(x, c) = \begin{cases} r_s, & \text{if } d_i(x, c) \text{ is small} \\ r_m, & \text{if } d_i(x, c) \text{ is medium} \\ r_l, & \text{if } d_i(x, c) \text{ is large} \end{cases} \tag{5}$$

$$\epsilon_i(x, f_i, c) = l(x, f_i)/r(x, c). \tag{6}$$

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 Sect. 8.

5.2 Angle Selection

In geo-indistinguishability, only the privacy of single location points is considered without real road maps, as shown in Fig. 11. The adversary can infer the actual

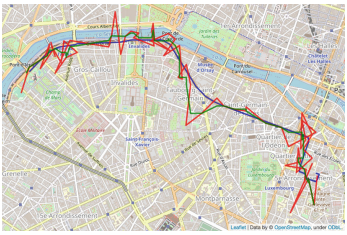

Fig. 1. An example output by PL_ϵ on a real map. Blue: actual trajectory, red: published, green: filtered. (Color figure online)

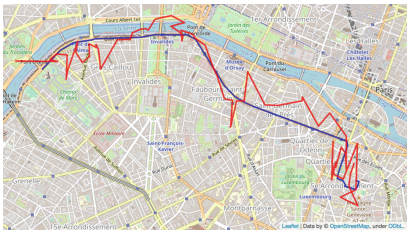

Fig. 2. An example output after the Angle Selection is applied. Blue: actual trajectory. Red: published. (Color figure online)

trajectory even if every location point is protected. With a median filter and real maps, the adversary can achieve a trajectory close to the actual one (as shown in Fig. 1). 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 [10] to add noise to the previous θ_0, and

$$\theta = \theta_0 + \mathcal{N}(\mu = 0, \sigma = \sqrt{2\ln(1.25/\delta)} \cdot \Delta_2/\epsilon_a). \tag{7}$$

We use ϵ_a as the privacy budget for the angle selection mechanism to distinguish it from the ϵ for geo-indistinguishability. With Eq. 7, we calculate the new angle θ and round it into the range $[0, 2\pi)$ (as Algorithm 2 in Appendix). The process mitigates the filtering attack by misleading the adversary to a wrong trajectory. We further analyze privacy protection in Sects. 7 and 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 [23]. By contrast, with the Gaussian noise, the scale is only $O(\sqrt{k})$.

Theorem 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 $\delta \in (0, 1]$ [20].*

In Theorem 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 [20].

6 Decentralized Location Sharing System

Initialization. With assumptions in Sect. 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 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 [22] as in Sect. 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. Protocol 1 establishes a shared key based on the international standard ISO/IEC 11770-2-6 [18]. After the shared key is derived, a (Truck, Sender, Receiver) triplet shares 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. With Protocol 3, Truck generates a random r and broadcasts the (R, P) pair. Sender and Receiver calculate the stealth address P' and find the match. The same record is shared with Receiver and Sender. In theory, we save half storage than PASTA [22].

Alice (Sender)		Bob (Receiver)
	$\xleftarrow{N_B}$	
	$\xrightarrow{\{N_A, N_B, ID_B, F_{AB1}, F_{AB2}\}_{K_{AB}}}$	
	$\xleftarrow{\{N_B, N_A, F_{BA1}, F_{BA2}\}_{K_{AB}}}$	
$k_{Aa} \leftarrow f(F_{AB1}, F_{BA1})$		$k_{Ba} \leftarrow f(F_{AB1}, F_{BA1})$
$k_{Ab} \leftarrow f(F_{AB2}, F_{BA2})$		$k_{Bb} \leftarrow f(F_{AB2}, F_{BA2})$

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.

Charlie (Truck)		Alice (or Bob)
ID_A, ID_B, p_{id}		
	$\xrightarrow{\text{Request } TK_{p_{id}A} \text{ and } TK_{p_{id}B}}$	
		$TK_{p_{id}A} \leftarrow H_s(p_{id}k_{Aa})G$
		$K_{Ab} \leftarrow k_{Ab}G$
	$\xleftarrow{TK_{p_{id}A}, K_{Ab}}$	
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)
$r \in_R Z_p, R \leftarrow rG$		
$P \leftarrow H_s(rTK_{p_{id}})G$		
$+K_b$		
	$\xrightarrow{\text{Broadcast } (R,P)}$	
		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

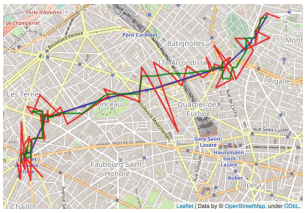

Fig. 3. Another example output by PL_ϵ. Blue: actual, red: published, green: filtered. (Color figure online)

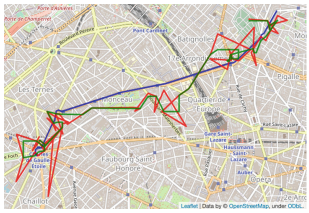

Fig. 4. Another example output with angle selection. Blue: actual, red: published, green: filtered. (Color figure online)

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.

7 Analysis

7.1 Security and Privacy Analysis

Location Perturbation. Section 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 [17] showing that a differentially private mechanism still suffers from attacks in real use cases. In this paper, with assumptions in Sect. 3, we consider privacy under real maps, showing that our proposed approach provides better privacy protection.

Figures 1, 2, 3 and 4 show example outputs from PL_ϵ and the proposed method for two different trajectories. In Figs. 1 and 3, 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 Fig. 4, 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 1, indicating that the proposed angle selection mechanism achieves stronger privacy guarantees than randomly selecting angles.

Lemma 1. *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. Figure 5 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

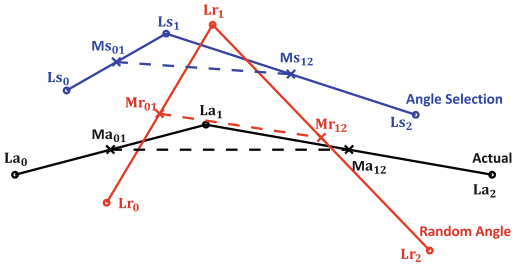

Fig. 5. An example trajectory with three location points.

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 of the guess distance $d \geq 0$ equal the published distance d_i is:

$$\frac{p(d = d_i)}{\int_0^\infty p(d)} = \frac{\frac{1}{b}\exp(-\frac{d_i}{b})}{\int_0^\infty \frac{1}{b}\exp(-\frac{d}{b})} = \frac{1}{b}\exp(-\frac{d_i}{b}) \tag{8}$$

where b is the scale and $|d|, |d_i| \geq 0$. Equation 8 shows that a smaller distance d_i means a higher probability of guessing the actual perturbation distance. The adversary can draw a circle with a radius equal 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 1, 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

$$\begin{aligned} 4 \cdot d_{01}^2 &= (x_{01} - (x_0 + x_1))^2 + (y_{01} - (y_0 + y_1))^2 \\ &= (r_0\cos(\theta_0) + r_1\cos(\theta_1))^2 + (r_0\sin(\theta_0) + r_1\sin(\theta_1))^2 \\ &= r_0^2 + r_1^2 + 2r_0r_1\cos(\theta_0 - \theta_1). \end{aligned} \tag{9}$$

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:

$$2v_d = \overrightarrow{Ma_{01}M_{01}} + \overrightarrow{Ma_{12}M_{12}} \tag{10}$$

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:

$$\begin{aligned}&(r_0\cos(\theta_0) + 2r_1\cos(\theta_1) + r_2\cos(\theta_2))^2 + (r_0\sin(\theta_0) + 2r_1\sin(\theta_1) + r_2\sin(\theta_2))^2\\ &= r_0^2 + 4r_1^2 + r_2^2 + 4r_0r_1\cos(\theta_0 - \theta_1) + 4r_1r_2\cos(\theta_1 - \theta_2) + 2r_0r_2\cos(\theta_0 - \theta_2).\end{aligned} \tag{11}$$

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 Eq. 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 Sect. 2, we assume location points are published every n minutes. Considering half-day 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 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 1 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 [12,21,35] 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.

Table 1. 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	Decryption	$\mathcal{O}(eN_P)$	–	–
Data Sharing	Encryption	–	$\mathcal{O}(eN_P)$	–
Smart Contract	Register	$\mathcal{O}(N_T)$	–	aN_T
Operation	Publish	$\mathcal{O}(N_L + N_P)$	–	$N_L e + N_P(k_{SE} + (p, r))$

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 [18] guarantees Alice and Bob can securely exchange key materials. The key derivation function PBKDF2 [24] 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 [14]: 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 A.1).

Lemma 2. *(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 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.

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 [25]. We summarize the computation complexity and on-chain storage in Table 1, 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 [22] (with proof in Appendix A.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 [31].

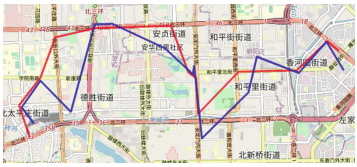

Fig. 6. Example output with $\epsilon = 0.01$.

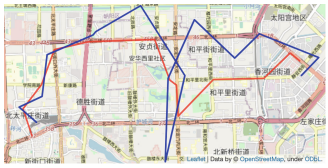

Fig. 7. Example output with $\epsilon = 0.005$.

Fig. 8. Example output with $\epsilon = 0.0025$.

Fig. 9. Example output with $\epsilon = 0.001$.

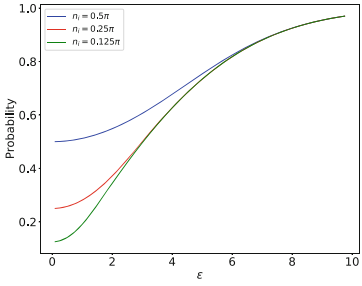

Fig. 10. Relation between the output probability and ϵ when the angle noise n_i is 0.125π, 0.25π and 0.5π.

8 Experimental Evaluation

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) [37] 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 (Eq. 12):

$$d = 2r \arcsin \sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}. \quad (12)$$

$\boldsymbol{\epsilon}$ *(for ϵ-geo-indistinguishability)* is selected by Eq. 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, we need to scale the privacy level l with the output results using different r. Table 2 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 6, 7, 8 and 9 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 Figs. 6 and 7. If $l = 1, r = 1000$, we have $\epsilon = 0.001$ as shown in Fig. 9 with much larger noise. After defining parameters l and r for the first time, the value of ϵ can be calculated in real uses.

$\boldsymbol{\epsilon_a}$ *(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 Fig. 10 using:

$$p(n_i) = \int_{-n_i}^{n_i} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} dx \Big/ \int_{-\pi}^{\pi} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} dx \quad (13)$$

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 Eq. 12, we evaluate the average distance (the distance 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 2. 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 6, 7, 8 and 9 and Table 2 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 min. However, for the adversary, Table 2 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 Fig. 9 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 m, which infers that the prediction error is within one minute. Figure 6 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.

Table 2. 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

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 [22] (2.864×10^6).

The scalability relies on the proof of work consensus model. For every second, Ethereum can process around 15 transactions [28], 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.

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 [22]. 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.

A Deferred Proofs and Figures

A.1 Proof for Lemma 2

Lemma 2. *(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:

$$\begin{aligned} P_1 - P_2 &= H_s(rTK_{p_{id1}})G + K_b - (H_s(rTK_{p_{id2}}) + K_b) \\ &= (H_s(rTK_{p_{id1}}) - H_s(rTK_{p_{id2}}))G \\ &= xG \text{ for some unknown x.} \end{aligned} \tag{14}$$

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:

$$\begin{aligned} P_1 - P_2 &= H_s(r_1TK_{p_{id1}})G + K_{b_1} - (H_s(r_2TK_{p_{id2}}) + K_{b_2}) \\ &= (H_s(r_1TK_{p_{id1}}) - H_s(r_2TK_{p_{id2}}) + K_{b_1} - K_{b_2})G \\ &= yG \quad \text{for any unknown y.} \end{aligned} \tag{15}$$

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

A.2 Proof of Lower Storage Cost

Lemma 3. *The proposed encryption method has lower storage costs than DECOUPLES [22].*

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:

$$\begin{aligned} S - S_{ECIES} &= N_L e + N_P(k_{SE} + (p, r)) - N_P(e + (p, r)) \\ &= (N_L - N_P)e + N_P(k_{SE} - e) \end{aligned} \tag{16}$$

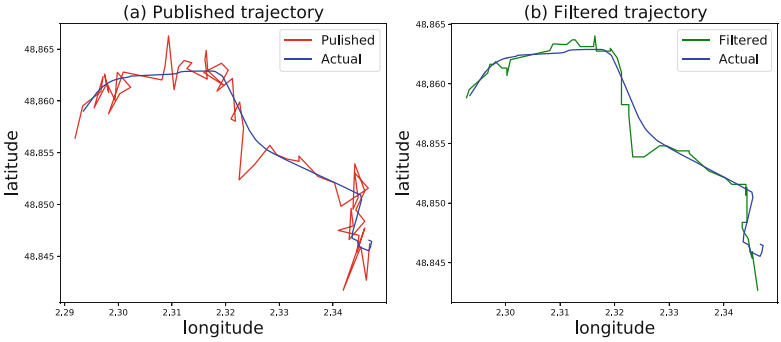

Fig. 11. An example output by PL_ϵ, and the filtered output of the sanitized trajectory. The blue line shows the actual trajectory, the red line shows the published trajectory by PL_ϵ, and the green line shows the filtered trajectory. (Color figure online)

Algorithm 2: AngleSelection

Input: Previous angle θ_0, privacy parameter ϵ_a
Output: Output perturbed angle θ
1: Calculate the new angle θ using Equation 7.
2: Round θ into the range $[0, 2\pi)$
3: **return** θ

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. Agatz, N.A.H., Campbell, A.M., Fleischmann, M., Savelsbergh, M.W.P.: Time slot management in attended home delivery. Transp. Sci. **45**(3), 435–449 (2011). https://doi.org/10.1287/trsc.1100.0346
2. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Sadeghi, A., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013, pp. 901–914. ACM (2013). https://doi.org/10.1145/2508859.2516735
3. Auditshipment: The true cost of package delivery delays (2021). https://www.auditshipment.com/blog/the-true-cost-of-package-delivery-delays/. Accessed 7 Nov 2021
4. Branch, A.E.: Global Supply Chain Management and International Logistics. Routledge, Abingdon (2008)
5. Brunswicker, S., Van de Vrande, V.: Exploring open innovation in small and medium-sized enterprises. New Front. Open Innov. **1**, 135–156 (2014)
6. DHL: Parcel delivery in real time (2021). https://www.dhl.de/en/privatkunden/pakete-empfangen/sendungen-verfolgen/live-tracking.html. Accessed 07 Jan 2022
7. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
9. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: Schulman, L.J. (ed.) STOC 2010, pp. 715–724. ACM (2010). https://doi.org/10.1145/1806689.1806787
10. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014). https://doi.org/10.1561/0400000042
11. Van den Engel, A., Prummel, E.: Organised theft of commercial vehicles and their loads in the European union. European Parliament. Directorate General Internal Policies of the Union. Policy Department Structural and Cohesion Policies. Transport and Tourism, Brussels (2007)
12. Fang, C., Chang, E.: Differential privacy with δ-neighbourhood for spatial and dynamic datasets. In: Moriai, S., Jaeger, T., Sakurai, K. (eds.) ASIA CCS 2014, pp. 159–170. ACM (2014). https://doi.org/10.1145/2590296.2590320
13. Grmiling, M.: How real time tracking can improve logistics (2021). https://www.hublock.io/how-real-time-tracking-can-improve-logistics/. Accessed 17 Nov 2021
14. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, Heidelberg (2006). https://doi.org/10.1007/b97644
15. Harnsamut, N., Natwichai, J., Riyana, S.: Privacy preservation for trajectory data publishing by look-up table generalization. In: Wang, J., Cong, G., Chen, J., Qi, J. (eds.) ADC 2018. LNCS, vol. 10837, pp. 15–27. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92013-9_2

16. Hayashida, S., Amagata, D., Hara, T., Xie, X.: Dummy generation based on user-movement estimation for location privacy protection. IEEE Access **6**, 22958–22969 (2018). https://doi.org/10.1109/ACCESS.2018.2829898
17. Hitaj, B., Ateniese, G., Pérez-Cruz, F.: Deep models under the GAN: information leakage from collaborative deep learning. In: Thuraisingham, B., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, pp. 603–618. ACM (2017). https://doi.org/10.1145/3133956.3134012
18. ISO: ISO/IEC 11770-2:2008, Information technology – Security techniques – Key Management – Part 2: Mechanisms using Symmetric Techniques (2009)
19. Jiang, H., Li, J., Zhao, P., Zeng, F., Xiao, Z., Iyengar, A.: Location privacy-preserving mechanisms in location-based services: a comprehensive survey. ACM Comput. Surv. **54**(1), 4:1–4:36 (2022). https://doi.org/10.1145/3423165
20. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. In: Bach, F.R., Blei, D.M. (eds.) ICML 2015. JMLR Workshop and Conference Proceedings, vol. 37, pp. 1376–1385. JMLR.org (2015). http://proceedings.mlr.press/v37/kairouz15.html
21. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. Proc. VLDB Endow. **7**(12), 1155–1166 (2014). https://doi.org/10.14778/2732977.2732989
22. Maouchi, M.E., Ersoy, O., Erkin, Z.: DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain. In: Hung, C., Papadopoulos, G.A. (eds.) SAC 2019, pp. 364–373. ACM (2019). https://doi.org/10.1145/3297280.3297318
23. McSherry, F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Çetintemel, U., Zdonik, S.B., Kossmann, D., Tatbul, N. (eds.) SIGMOD 2009, pp. 19–30. ACM (2009). https://doi.org/10.1145/1559845.1559850
24. Moriarty, K.M., Kaliski, B., Rusch, A.: PKCS #5: password-based cryptography specification version 2.1. RFC **8018**, 1–40 (2017). https://doi.org/10.17487/RFC8018
25. Saberi, S., Kouhizadeh, M., Sarkis, J., Shen, L.: Blockchain technology and its relationships to sustainable supply chain management. Int. J. Prod. Res. **57**(7), 2117–2135 (2019). https://doi.org/10.1080/00207543.2018.1533261
26. Sahai, S., Singh, N., Dayama, P.: Enabling privacy and traceability in supply chains using blockchain and zero knowledge proofs. In: Blockchain 2020, pp. 134–143. IEEE (2020). https://doi.org/10.1109/Blockchain50366.2020.00024
27. Savona, E.U.: Organised property crime in the EU. European Parliament. Directorate General for Internal Policies. Policy Department for Citizens' Rights and Constitutional Affairs (2020)
28. Seres, I.A., Nagy, D.A., Buckland, C., Burcsi, P.: Mixeth: efficient, trustless coin mixing service for ethereum. Cryptology ePrint Archive, Report 2019/341 (2019)
29. Sezer, B.B., Topal, S., Nuriyev, U.: An auditability, transparent, and privacy-preserving for supply chain traceability based on blockchain. CoRR abs/2103.10519 (2021). https://arxiv.org/abs/2103.10519
30. Shoup, V.: A proposal for an ISO standard for public key encryption. IACR Cryptol. ePrint Arch, p. 112 (2001). http://eprint.iacr.org/2001/112
31. Singh, A., Click, K., Parizi, R.M., Zhang, Q., Dehghantanha, A., Choo, K.R.: Sidechain technologies in blockchain networks: an examination and state-of-the-art review. J. Netw. Comput. Appl. **149** (2020). https://doi.org/10.1016/j.jnca.2019.102471

32. Terrovitis, M., Poulis, G., Mamoulis, N., Skiadopoulos, S.: Local suppression and splitting techniques for privacy preserving publication of trajectories. IEEE Trans. Knowl. Data Eng. **29**(7), 1466–1479 (2017). https://doi.org/10.1109/TKDE.2017.2675420
33. Wang, H., Xu, Z.: CTS-DP: publishing correlated time-series data via differential privacy. Knowl. Based Syst. **122**, 167–179 (2017). https://doi.org/10.1016/j.knosys.2017.02.004
34. Wong, L., Leong, L., Hew, J., Tan, G.W., Ooi, K.: Time to seize the digital evolution: adoption of blockchain in operations and supply chain management among malaysian smes. Int. J. Inf. Manag. **52**, 101997 (2020). https://doi.org/10.1016/j.ijinfomgt.2019.08.005
35. Xiao, Y., Xiong, L.: Protecting locations with differential privacy under temporal correlations. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015, pp. 1298–1309. ACM (2015). https://doi.org/10.1145/2810103.2813640
36. Xiong, P., Zhu, T., Pan, L., Niu, W., Li, G.: Privacy preserving in location data release: a differential privacy approach. In: Pham, D.-N., Park, S.-B. (eds.) PRICAI 2014. LNCS (LNAI), vol. 8862, pp. 183–195. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13560-1_15
37. Zheng, Y., Zhang, L., Xie, X., Ma, W.: Mining interesting locations and travel sequences from GPS trajectories. In: Quemada, J., León, G., Maarek, Y.S., Nejdl, W. (eds.) WWW 2009, pp. 791–800. ACM (2009). https://doi.org/10.1145/1526709.1526816