Secure Trajectory Publication in Untrusted Environments: A Federated Analytics Approach

Zibo Wang, Yifei Zhu, *Member, IEEE,* Dan Wang, *Senior Member, IEEE,* and Zhu Han, *Fellow, IEEE*

Abstract—The increasing awareness of privacy and the adoption of data regulations challenge the traditional trajectory publication framework in which a trusted server has access to the raw data from mobile clients. In the new untrusted environment, the clients call for much stronger data privacy preservation locally without sharing their raw data. Based on the emerging paradigm of federated analytics, we propose a Federated Analytics-based Secure Trajectory PUBlication (FASTPub) mechanism to operate in such untrusted environments. Compared with existing local differential privacy (LDP) methods, FASTPub guarantees LDP and loss-bounded k-anonymity simultaneously with greatly improved data utility. Specifically, FASTPub works interactively between the server and clients and iteratively builds up the trajectory without exposing raw data. Sampled clients only respond to selected trajectory fragments with randomized answers to preserve privacy as much as possible. The server then intelligently aggregates these randomized responses leveraging the intrinsic Apriori property and a Markov independent assumption of trajectory data to guide further iterations. Extensive experiments on synthetic and real-world datasets on two downstream tasks demonstrate that FASTPub gains a remarkably improved data utility compared to the existing state-of-the-art solutions.

Index Terms—federated analytics, trajectory publication, local differential privacy, k-anonymity, collaborative computing

1 Introduction

The advances in location-acquisition and mobile computing techniques have generated massive trajectory data, represented as sequences of chronologically ordered geometric or virtual locations (*e.g.* geographical locations or website browsing history) from moving objects. In 2020, an estimated 8 billion mobile devices received their location information from the Global Navigation Satellite System (GNSS) [1]. From these trajectory data, valuable knowledge can be derived in both the macroscopic perspective, *e.g.* daily/annual human mobility trends analysis [2], and the microscopic perspective, *e.g.* road routing [3]. These mined information further powers the growth of various applications and services, like vehicular networks [4], location-based services [5], and smart city [6].

As the cornerstone of all trajectory data mining tasks, the trajectory data have to be gathered first and form a database, a procedure usually called trajectory publication [7]. Traditionally, a server of a trusted third party first gathers the trajectory data from mobile devices, and then releases it to the data analysts for research purposes after enforcing certain privacy constraints. For example, Nokia gathers trajectory data from 185 volunteers from Lausanne, Switzerland, and releases the Mobile Data Challenge (MBC) dataset with k-anonymity enforced centrally for privacy preservation [8]; Microsoft gathers the trajectory data of taxis in Beijing, China, and releases it as an anonymous T-Drive dataset [9]. In the traditional data publication framework, the privacy issue is only considered when the trajectory database is released to the data analyst, while the procedure of acquiring data from the clients[1] is done straightforwardly with little consideration of local privacy [10], [11], [12], [13].

Nowadays, with the increasing awareness of privacy preservation, laws and regulations are established to limit the collection of clients' data, such as the EU General Data Protection Regulation (GDPR) [14]. The trend fundamentally changes the underlying assumption in the classical trajectory publication framework: there will be no such admitted "trusted server" having access to raw data in the clients. In the resulting *untrusted* environment, the clients worry about the danger caused by uploading *sensitive information* to prevent potential privacy attacks, such as re-identification. As a result, they strive to avoid direct transmission of raw data, and request local privacy preservation.

k-anonymity and differential privacy (DP) are two de facto industrial and academic standards in the field of privacy preservation. k-anonymity [15] enables an individual to "hide in the crowd" of companions with similar exposed data, while local differential privacy (LDP) [16], the local privacy scheme of DP, regulates the uploads to reduce individual biases via randomization. These two criteria become equally important in the emerging untrusted data environments. On one hand, LDP reduces the identity of

- *The work of Yifei Zhu was funded by the SJTU Explore-X grant. The work of Dan Wang was supported in part by GRF under Grants 15210119, 15209220, and 15200321, in part by ITF-ITSP under Grant ITS/070/19FP, in part by CRF under Grants C5026-18 G and C5018-20 G, in part by PolyU 1-ZVPZ, in part by Huawei Collaborative Project. The work of Zhu Han was partially supported by NSF under Grants CNS-2107216 and CNS-2128368.*
- *The corresponding author is Yifei Zhu.*

- *Z. Wang and Y. Zhu are with UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai, China*
  E-mail: {wangzibo, yifei.zhu}@sjtu.edu.cn
- *D. Wang is with Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.*
  E-mail: csdwang@comp.polyu.edu.hk
- *Z. Han is with Department of Electrical and Computer Engineering, University of Houston, Houston, USA.*
  E-mail: zhan2@uh.edu

[1] We use the term *client* to indicate the distributed trajectory data owner in this paper.

each client by forcing their outputs to be closed to each other. On the other hand, k-anonymity protects the users with "rare" data from being identified.

However, both criteria are difficult to be realized and ineffective to be deployed in the emerging untrusted trajectory scenario. For k-anonymity, it requires every uploaded information to have at least $k-1$ similar companions, and counting the number of duplicates requires gathering raw data from other clients. In other words, the central privacy-oriented k-anonymity cannot be easily adjusted with only local information. Meanwhile, since the space of trajectory is extremely large, realizing LDP via naïve perturbation introduces significant noise on the raw trajectory data, greatly harming the publication quality. Consequently, two criteria all require global knowledge from other clients, which conflicts with the requirement of data staying local in an untrusted environment.

Since 2020, federated analytics (FA), as a sequel to the widely studied federated learning (FL), has started to attract both academia and industry's attention [17], [18], [19], [20]. This new data analytics paradigm studies the scenarios where a central server hosts a data analytics task with many distributed clients holding private data locally. Instead of sending the raw data directly to the server, the clients perform a part of the data mining procedure solely with their local data, generate indirect and insensitive insights based on their local data, and upload these insights to the server. After that, the server aggregates the insights in a non-trivial way, and generates new guides for the clients to push forward the data analytics task. In FL, the host task is to train a neural network, so the local insight derivation is essentially neural network training (gradient descent), and the aggregation is gradient averaging. On the contrary, FA focuses on data science tasks that cannot be solved by neural networks, so it significantly differs from FL in the form of local insights derivation and central aggregation. Therefore, we choose FA to accomplish the non-training-based trajectory publication task in an untrusted environment.

However, how to achieve trajectory publication under the paradigm of FA so that local privacy can be preserved by both k-anonymity and LDP has never been studied before. In this paper, following the FA paradigm, we propose a Federated Analytics-based Secure Trajectory PUBlication, or FASTPub[2], a mechanism to publish non-trivially long *trajectory fragments* in untrusted environments. Essentially, FASUPub builds a trajectory database with a predefined length by gathering fragments (continuous subsequences) of trajectory from the clients[3]. FASTPub is interactive and iterative, where the clients start by publishing the shortest fragments of trajectory with length 1. Only after the server infers out the valid longer fragment candidates based on the aggregated responses to the shorter ones, it will then query a new group of randomly selected clients on their existence. The publication process continues until the predefined length is reached. Instead of directly adding noise to a long trajectory chain, the clients only need to answer yes/no for a limited number of candidate trajectory fragments generated by the server. On the other hand, by iteratively increasing the fragment length, FASTPub successfully bounds the expectation of anonymity loss for k-anonymity, which provides valid privacy preservation in untrusted environments. FASTPub simultaneously satisfies the two criteria, which provides greater privacy preservation than satisfying either of them solely.

TABLE 1
Comparison between data publication solutions. We analyze their ability in trajectory publication tasks, data utility as low (+), moderate (++), and high (+++), and privacy guarantees.

	Trajectory data	Utility	Privacy
RAPPOR [21]	X	+++	Local DP
[10], [11], [22]	✓	+++	Central DP
[12], [13]	✓	+++	Central k-anon.
SFP [23]	✓	+	Local DP
TrieHH [19]	✓	++	Central DP[4]
FASTPub	✓	+++	Local DP and local k-anon.

In summary, our contributions are:

1) We present the first work to realize secure trajectory data publication in the untrusted environment following the paradigm of FA.
2) We disaggregate the publication services into iteratively publishing trajectory fragments, and judiciously leverage the Apriori property and Markov independent assumption of the trajectory data to realize LDP with greatly improved utility.
3) To further protect privacy, we extend the central k-anonymity definition to be viable in the local privacy scenario. The extended loss-bounded k-anonymity is satisfied in FASTPub via the enforcement of (k, ξ)-anonymity (satisfy k-anonymity with confidence $1 - \xi$) on the candidate fragments.
4) We theoretically prove that FASTPub enforces LDP and loss-bounded k-anonymity, and calculate the support count threshold of candidates to satisfy (k, ξ)-anonymity. These expressions reveal the reliability of FASTPub in local privacy preservation.
5) Extensive experiments on real-world and synthetic datasets show that, compared to two state-of-the-art benchmarks, the database generated by FASTPub gains $2.5 \sim 6.1$ times higher F1 score in frequent pattern mining task, and $85.6\% \sim 87.2\%$ reduced relative error in count query task.

The paper is organized as follows: the related work is surveyed in Section 2. Preliminaries and motivation are in Section 3. We present the system model and problem formulation in Section 4. Section 5 includes the design details and the corresponding theoretical analysis of FASTPub. Evaluations are illustrated in Section 6, and a conclusion is finally drawn in Section 7.

2 Related Work

2.1 Privacy-preserving Trajectory Publication

Extensive studies have been done to protect the privacy of trajectory data. To verify the effectiveness, the proposed

[2] Our implementation is available in https://github.com/inslab-ji/FASTPub.

[3] The necessity of publishing trajectory fragments instead of raw trajectories is demonstrated in Section 4.1.

[4] The authors of [19] claim that TrieHH provides local privacy preservation to some extent by limiting the raw data exposure.

methods mainly target privacy criteria including central differential privacy [10], [11], [22] and k-anonymity [12], [13], which work in the trusted data environment with a data aggregator accessing all the raw data. Consequently, the central privacy scheme devised there cannot meet the requirement of user-level privacy in our studied untrusted data environment. Numerous research efforts have been proposed in preserving local privacy. The randomized response is adapted by various real-world deployments to meet LDP, a typical local privacy preservation criterion with a rigorous mathematical definition [21], [23], [24], [25]. However, they suffer from great utility loss if naïvely adding significant noise to the raw data. In addition, researchers have also tried to design local privacy schemes based on k-anonymity [26], [27]. Unfortunately, the aforementioned local privacy schemes are either unfeasible or ineffective to be deployed in the trajectory scenario. Notably, as the basis of trajectory data, the location data have been given many solutions for privacy preservation [28], [29], [30]. However, the trajectory scenario is much more complicated than the former due to the high dimensionality of sequential trajectory data. In this paper, we propose FASTPub, the first work to combine LDP and local variation of k-anonymity, which provides thorough protection against multiple attack strategies. It also achieves a greatly improved data utility in the untrusted publication environments, handling the difficult trajectory data structure. A comparison between FASTPub and existing solutions is presented in Table 1.

2.2 Federated Analytics

Different from FL [31], where a neural network is collaboratively trained, FA focuses on non-training oriented data mining tasks. Given the wide spectrum of data mining tasks, there is still no universal framework for FA at the current nascent state. Several pioneering works have already focused on devising FA mechanisms to help FL or conduct specific analytic tasks in the decentralized data environment. In [18], the concept of FA is firstly introduced, and its applications on FL model evaluation and song recognition are presented. In [20], an FA mechanism is proposed to measure the severity of data heterogeneity in edge devices to help other federated optimization tasks. In [32], an FA scheme is designed to reveal single value data by uploading one bit from its local value. To the best of our knowledge, there is still no FA work specifically designed for trajectory publication scenarios. The most related work is the federated frequent word analytic mechanism [19], because the frequent word can be treated as a type of sequence data. However, it settles with central DP, instead of LDP, for utility improvement. On the contrary, our solution successfully achieves both stricter LDP and higher data utility simultaneously.

3 Preliminaries and Motivation

In this section, we first introduce the necessity of having k-anonymity and LDP satisfied simultaneously in the emerging untrusted environments in Section 3.1. Then, the two criteria in our locally untrusted settings are introduced in Sections 3.2 and 3.3, respectively. After that, we present the motivation for FASTPub based on reviewing the limits of existing methods in Section 3.4.

3.1 Necessity of Having Two Criteria

There exists an intensive debate on the effectiveness of LDP and k-anonymity [22], [33], [34]. As a newly introduced concept, DP is believed by many researchers to provide stronger privacy preservation than k-anonymity [22], [33]. On the other hand, in [34], some scenarios where DP does no longer work are identified. Neither of the two criteria can thoroughly outperform the opposite since they tackle different attack strategies. LDP assumes that a re-ID attacker has full knowledge of the prior distribution of client data, *i.e.*, the attacker has a complete but anonymous collection of client data, and it wants to match the clients with the records. LDP protects the client data from being inferred by the attacker with a rigorous guarantee.

However, the LDP privatized data can still be treated as a fingerprint of the client, which may harm data privacy [34]. As a result, the attacker can trace the client after the publication even if the attacker does not have any prior knowledge. The risk is comprehensively reasonable in the trajectory publication scenario because 1) the sparse trajectory data is naturally suitable for fingerprints, and 2) the clients in trajectory publication are usually mobile, and tracing the clients with their fingerprints introduces trajectory leakage. For example, when an attacker traces a smartphone via a cellular network by seeking a particular historical trajectory publication record, it can reconstruct the trajectory even if the client would like to be anonymous. k-anonymity eliminates this risk by guaranteeing that the privatized upload is not a unique fingerprint.

In conclusion, simultaneously meeting the two criteria provides better privacy preservation than satisfying any of them solely. In fact, researchers in [35], [36], [37] have proposed some pioneering works to blend DP and k-anonymity in the central privacy setting with a trusted aggregator.

3.2 Local Privacy Preservation of k-anonymity

Introduced in [15], k-anonymity is an effective tool in preventing individuals from the re-identification attack. Consider the environment with N clients, denoted by $c_1, c_2, ..., c_N$. Each client uploads some data to a centralized server. In k-anonymity, the uploaded data from one client c_i consists of several attributes named *quasi-identifier*, which can be used by a potential adversary to identify one individual, and *sensitive attributes*, which people do not want others to know about. Denote the quasi-identifier from c_i as d_i, and the set of d_i from all clients as $\mathcal{Q}$. We define a support count function $\mathcal{S}(\cdot)$ for any d_i as follows:

$$\mathcal{S}(d_i) = \sum_{d_j \in \mathcal{Q}} \mathbb{I}(d_i = d_j), \tag{1}$$

where $\mathbb{I}$ is the indicator function, returning 1 when d_i is the same as d_j from any other client (or itself), and 0 otherwise.

Definition 1 (k-anonymity)**.** *A data publication algorithm satisfies k-anonymity when*

$$\mathcal{S}(d) \geq k, \ \forall d \in \mathcal{P}, \tag{2}$$

where $\mathcal{P}$ are all the published data of the algorithm.

k-anonymity forces any published data to have at least $k-1$ companions with the same quasi-identifiers. It provides privacy preservation following the idea of "hiding in the crowd". With k-anonymity, potential adversaries can no longer match a person with a specific quasi-identifier and infer his/her sensitive attributes, because he/she is hiding in the crowd of k people with different sensitive attributes.

Originally, k-anonymity serves as a central privacy scheme, which protects clients' privacy when the server releases the aggregated data to the public. At the moment a datum is published by a client, neither the client nor the server knows whether it satisfies k-anonymity, which prevents k-anonymity from providing local privacy preservation. Existing works try to gain local privacy preservation via k-anonymity, however, they rely on secure multi-party computation infrastructure and cannot be applied to the trajectory data [26], [27].

In FASTPub, we investigate the characteristic of trajectory data and extend the centrally defined k-anonymity to handle local privacy scenarios, with carefully designed *anonymity loss*. The anonymity loss measures the potential privacy leakage of uploading any trajectory chain. The idea of loss-bounded k-anonymity is simple: even if a trajectory chain may not satisfy k-anonymity, we try to guarantee that a long continuous subsequence of it satisfies k-anonymity, which leads to comparatively small privacy leakage. Realizing loss-bounded k-anonymity is feasible in FASTPub thanks to its interactive scheme between server and clients, and the detailed definitions of anonymity loss and loss-bounded k-anonymity are present in Section 4.2.

3.3 Local Differential Privacy

As the origin of LDP, DP is a set of criteria for preventing individuals from being identified in a database [38]. The earliest and most popular version of DP is central differential privacy (CDP). However, the functionality of CDP requires a trusted aggregator having access to all raw data [10], [11]. As a result, CDP does not work in an untrusted environment, because there is no such aggregator trusted by all clients, and raw data uploading is not allowed. Therefore, we used the local version of DP, named local differential privacy (LDP) [16], whose definition is defined as follows[5].

Definition 2 (local differential privacy)**.** *A randomized algorithm $\mathcal{M}$ satisfies ϵ-LDP when for any two possible records a and b, and any possible output y*

$$\mathbb{P}(\mathcal{M}(a)=y) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(b)=y),\ \forall y \in \mathrm{Range}(\mathcal{M}). \tag{3}$$

To realize LDP, adding noise (perturbation) to the outputs is necessary.

3.4 Failure of Classical Methods

In this part, we consider several classical methods in enforcing k-anonymity or LDP, and analyze why they do not work in the trajectory publication in untrusted environments.

k-anonymity naturally faces some troubles when tackling trajectory data. A trajectory has to have at least $k-1$ companions with the same length, locations, and ordering of locations to satisfy k-anonymity, which is rare in the real world and thus becomes impractical to enforce in the trajectory settings. Therefore, existing methods tackle this problem by regulating the trajectories from different clients to be their common subsequence [13], or merging neighboring locations to a common one [12]. However, these approaches naturally require an aggregator, and only provide central privacy preservation. For example, a popular design of enforcing k-anonymity is to group the clients into clusters with a size of at least k, and replace the quasi-identifiers of each client with a common one of each cluster. It naturally requires the clients to share the raw data to calculate the similarity of the quasi-identifiers between clients, which is not allowed in untrusted environments.

On the other hand, the enforcement of LDP naturally requires adding noise (perturbation), which faces challenges when being applied to trajectory data with high dimensionality and an extremely large domain. We consider a conservative example that there are 100 possible locations, and the length of trajectory is limited to 3. The possible trajectories (outputs) are defined by the 3-length permutations (with replacement) of the 100 locations, with a large size of 10^6. Adding noise to provide each of the outputs with some possibility will ruin the utility. Even worse, for the raw data collected by edge devices, the possible length of trajectory might be different and unlimited, which further increases the possible domain to infinite.

4 System Model and Problem Statement

In this section, we first present the system model in Section 4.1. Then, we formally present our studied trajectory publication problem with loss-bounded k-anonymity and LDP satisfied in Section 4.2. After that, we discuss the local privacy preservation ability of loss-bounded k-anonymity in Section 4.3.

4.1 System Model

In trajectory publication applications, each client owns a private trajectory in the form of a sequence of locations. Coordinates with continuous values can also be adapted into our setting by discretization with a grid. An example of trajectory t can be expressed as:

$$t := a_1^{<t>} \to a_2^{<t>} \to \cdots \to a_{|t|}^{<t>}, \tag{4}$$

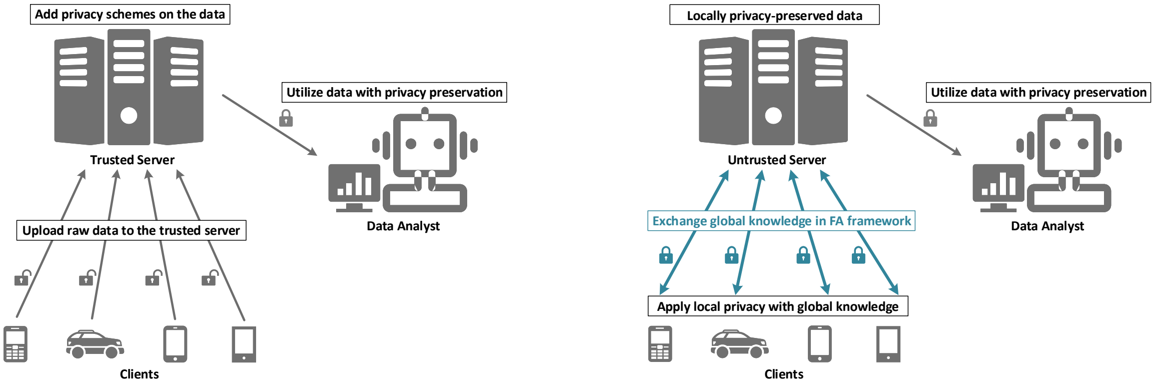
where $a_i^{<t>}$ indicates the i-th location of the trajectory t, and $|t|$ is the length (number of locations) of t.

For a trajectory t, we further define its *trajectory fragments* as the continuous subsequences of the original trajectory.

Definition 3 (l-length trajectory fragment)**.** *A trajectory fragment (or fragment) is a continuous subsequence of the original trajectory or another fragment. We define $t[p:q]$ as the fragment of t between the p-th and the q-th locations, i.e.,*

$$t[p:q] := a_p^{<t>} \to a_{p+1}^{<t>} \to \cdots \to a_q^{<t>}, \tag{5}$$

[5] ϵ controls the strength of privacy preservation, where lower ϵ indicates stronger privacy preservation.


Add privacy schemes on the data
Trusted Server
Utilize data with privacy preservation
Data Analyst
Upload raw data to the trusted server
Clients
Locally privacy-preserved data
Untrusted Server
Utilize data with privacy preservation
Data Analyst
Exchange global knowledge in FA framework
Apply local privacy with global knowledge
Clients

(a) Traditional trajectory publication

(b) FA assisted trajectory publication

Fig. 1. Left: traditional trajectory publication framework with a trusted aggregator; Right: FA assisted trajectory publication with local privacy preservation and data cleaning in the untrusted environment.

where $1 \leq p \leq q \leq l_t$. The length of a trajectory fragment can be derived by $l = p - q + 1$. For simplicity, we use l-fragment to denote a fragment with length l.

Note that we can freely treat the input t in (5) as an original trajectory or another trajectory fragment, because a fragment is in the same form as the original trajectory, *i.e.*, a trajectory fragment can have trajectory fragments of shorter lengths.

Threat model: The untrusted environment introduces a novel threat model of trajectory publication tasks. In traditional trajectory publication settings, as demonstrated in Fig. 1(a), the adversary can only contact data published by the trusted server. Therefore, clients just need to submit the trajectory data to the trusted server with no privacy consideration. It is the server's responsibility to aggregate the local data and provide central privacy schemes. In the untrusted setting, the adversary is able to read the data stored in the server. As a result, the previous approach will directly expose the raw trajectory. We assume an honest but curious adversary, which faithfully executes the trajectory publication procedure, but tries to learn about the clients' privacy based on the information exposed to the server. The adversary breaches the clients' privacy with two strategies. First, it tries to recover the clients' raw trajectories from their uploads. Second, it leverages the client's upload history as fingerprints, and uses the uploads to break the anonymity of clients. After that, it can trace a client from the crowd by checking its upload history (we assume the adversary may arbitrarily check the publication history of a client).

In this paper, we publish trajectory fragments instead of raw trajectories. It is necessary for the feasibility of privacy preservation, both k-anonymity and LDP. If we require k-anonymity on published data, choosing raw trajectory for publication often results in publishing nothing, because there might be no k clients holding an identical trajectory chain. In addition, satisfying LDP on the arbitrary-length raw trajectory is, if not impossible, extremely difficult, because the output space becomes infinite. As a result, a majority of trajectory publication solutions trim the raw trajectory like us to satisfy centralized k-anonymity [12], [13] or LDP [23]. The trimmed trajectory fragment data are also helpful in many data analytics applications [10], [11].

In FASTPub, we leverage the FA paradigm to perform knowledge exchange between server and clients with privacy preserved. As is demonstrated in Fig. 1(b), in each round, the server provides directions based on the global knowledge to a randomly selected subset of the clients. The directions are helpful for the participating clients to distill their local data. By receiving the distilled information, the server updates the global knowledge, and starts a new round. The local privacy is preserved because LDP is enforced and anonymity loss is bounded for any client upload. In addition, we restrict each client to only participate in one round, we decrease the information leakage for each client.

4.2 Problem Statement

As is mentioned in Section 3.2, k-anonymity cannot provide local privacy solely. Therefore, we define *anonymity loss* $\tau(f)$ as a measure of privacy leakage due to the failure of k-anonymity on trajectory chain f.

Definition 4 (anonymity loss)**.** *Denote f as any trajectory chain, its anonymity loss $\tau(f)$ is the difference of the lengths between f and its longest fragments that satisfies k-anonymity (or 0 if there is no such fragment), i.e.,*

$$\tau(f) = |f| - \max_{0 \leq p \leq q \leq |f|} |f[p:q]|, \tag{6}$$

$$s.t. \quad \mathcal{S}(f[p:q]) \geq k, \tag{7}$$

where $|f|$ describes the length of f.

The rationale for anonymity loss is that, if a major part of f has satisfied k-anonymity, even if f itself fails to satisfy it, the potential privacy leakage, which is measured by anonymity loss, tends to be lower. By bounding anonymity loss, the power of k-anonymity is leveraged to preserve local privacy.

We then formulate our problem mathematically. Denote the set of clients as $\mathcal{N}$; $\mathcal{M}$ is the publication mechanism in the clients; a and b are any theoretically possible trajectory stored in a client; Denote l_{max} as the target fragment length, $\mathcal{X}_c$ as the set of all uploaded l_{max}-fragments from the client c, and $\mathcal{P}_c$ as the set of those being admitted by the server. $\mathcal{T}$ is the anonymity loss limit.

We aim at designing mechanisms to elicit fragments with the targeted length as many as possible under the satisfaction of LDP and limitation of the expectation of anonymity loss, namely,

$$\max \sum_{c\in\mathcal{N}} |\mathcal{P}_c|, \tag{8}$$

$$s.t.\quad \mathbb{P}(\mathcal{M}(a)=y) < e^{\epsilon}\mathbb{P}(\mathcal{M}(b)=y),\ \forall y \in \mathrm{Range}(\mathcal{M}), \tag{9}$$

$$\mathbb{E}\Big(\sum_{f\in\mathcal{X}_c} \tau(f)\Big) \le \mathcal{T},\ \forall c \in \mathcal{N}, \tag{10}$$

where (9) describes the satisfaction of LDP, and (10) describes the satisfaction of loss-bounded k-anonymity that bounds the expectation of anonymity loss.

4.3 Discussion

FASTPub waives the satisfaction of vanilla k-anonymity, *i.e.*, the clients are not provided any guarantee that their published data can satisfy k-anonymity with any confidence level. However, the weaker loss-bounded k-anonymity still preserves local privacy to some extent. We consider a client who will publish a trajectory $a \to b \to c \to d$. Loss-bounded k-anonymity gives a guarantee to the client beforehand that both $a \to b \to c$ and $b \to c \to d$ have satisfied k-anonymity with sufficient confidence. As a result, the client can trust that the attacker is unlikely identified the client by the published trajectory data. In Section 3.1, we claim that k-anonymity can prevent the uploaded data from being treated as fingerprints. In FASTPub, the loss-bounded k-anonymity provides protection with the assistance of LDP, because the clients upload binary responses with randomization, which is not unique enough to be used as fingerprints. On the contrary, LDP-only schemes like SFP [23] require clients to upload complex data structures, which have a high risk of fingerprint tracing.

5 FASTPub: Design and Analysis

In this part, we first give an overview of the FASTPub framework in Section 5.1. The three components of FASTPub are then demonstrated in detail in Sections 5.2, 5.3, and 5.4, respectively. After that, we provide the theoretical analysis for satisfying the design goals in Section 5.5.

5.1 FASTPub Overview

The main idea of FASTPub is simple: the server infers candidates of longer fragments based on the shorter ones, and the clients generate binary responses on those candidates, which decreases the noise added and preserves more data utility when applying LDP. Specifically, FASTPub functions iteratively, and the length of published fragments increases by 1 in each round. Each round of FASTPub has three phases: *candidate generation, client response, and fragment filtering*. In the candidate generation phase, the server uses the Apriori property to generate candidates based on the results in the last previous round, or independently generate some candidates in the first round. If the length of candidates is sufficiently long, the server then cleans the candidates based on the Markov independent assumption. In the client response phase, the server samples some participating clients that have not participated before, and lets them respond about the existence of the candidates, where LDP is enforced on the yes/no responses. In the candidate filtering phase, the server only reserves the candidate with sufficient support counts, so that the reserved fragments are valid for the Apriori property in the next round, and guarantee the loss-bounded k-anonymity of the candidates in the next round. Fig. 2 gives an overview of FASTPub.

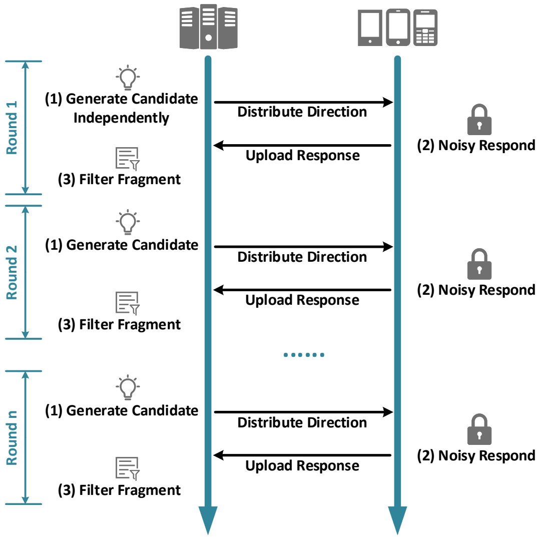


Fig. 2. An overview of FASTPub.

FASTPub is an interactive and iterative mechanism. Its interactive design enables information flow both from server to client and from client to server. In possible non-interactive trajectory publication solutions, valuable information is transmitted from client to server only. The interactivity of FASTPub is realized by the query-response scheme, which significantly reduces the utility loss under the guidance of the server and satisfies the LDP at the same time. In the meantime, note that such a design also introduces extra communication in downloading, and usually introduces a heavier computation load. The iterative FASTPub divides the whole procedure into rounds, and lets each client participate in one of the rounds. The interactive query-response scheme is benefited from the “iterativity” by increasing the quality of queries based on previous responses. However, an iterative scheme enforces stricter requirements on the participating clients, that they should participate in a smaller time window.

5.2 Candidate Generation and Cleaning

In the first round, the server independently generates candidates by enumerating all possible 1-fragments. In the later rounds, the candidates are generated based on the shorter admitted ones, whose procedure consists of two steps: Apriori property-based candidate generation and Markov independent assumption-based candidate removal.

Apriori property-based candidate generation: Firstly proposed in [39], the Apriori algorithm is simple but efficient in association rule mining. It is based on the Apriori property: for any itemset I, a prerequisite of that I has a support count not less than k is that all subsets of I have a support count not less than k. Although the Apriori property mostly focuses on the unordered itemsets, we reveal that it can also be adopted in the sequential trajectory data.

Theorem 1 (Apriori property of trajectory data)**.** *For any trajectory l-fragment f where $l \geq 2$, a necessary requirement of that f has support count not less than k is that its two $(l-1)$-fragments of f have support count not less than k, i.e.,*

$$\mathcal{S}(f) \geq k \implies \mathcal{S}(f[1:l-1]) \geq k \wedge \mathcal{S}(f[2:l]) \geq k. \quad (11)$$

Since a prerequisite for any fragment to satisfy k-anonymity has been provided by the Apriori property in Theorem 1, we can construct our secure trajectory publication algorithm via utilizing the theorem inversely, *i.e.*, generate a candidate set of l-fragments based on the $(l-1)$-fragments that have satisfied k-anonymity.

Denote the set of all l-fragments as Ω_l, and the set of l-fragments satisfying k-anonymity as $\mathcal{P}_l$. We can generate a candidates set $\mathcal{C}_{l+1}$ of $(l+1)$-fragments that are possible to satisfy k-anonymity:

$$\mathcal{C}_l = \{\forall f \in \Omega_l \mid f[1:l-1] \in \mathcal{P}_{l-1} \wedge f[2:l] \in \mathcal{P}_{l-1}\}. \quad (12)$$

For example, if the two fragments $a_1 \to a_2 \to a_3$ and $a_2 \to a_3 \to a_4$ are present in $\mathcal{P}_3$, we can then generate a candidate 4-fragment $a_1 \to a_2 \to a_3 \to a_4$.

Note that we only restrict the two $(l-1)$-fragments instead of all fragments of f, because the Apriori property can be recursively applied to the two shorter fragments, and derive the same results for all fragments of f. Based on the Apriori property and the $(l-1)$-fragments satisfying k-anonymity, we can generate a set of candidates of l-fragments, which are feasible to satisfy k-anonymity.

Markov independent assumption-based candidate cleaning: The Apriori property discovers the possible l-fragments based on knowledge of the shorter ones. However, it reserves every candidate which is theoretically possible to satisfy k-anonymity, whose number is likely to be large. A large number of candidates, including some with limited likelihood to satisfy k-anonymity, do not only require more participating clients for verification, but also increase the potential anonymity loss. Therefore, we clean these low-quality candidates based on the Markov independent assumption. This approach is inspired by the authors of [11], where they use it to construct a synthetic database based on the fragments.

The Markov independent assumption assumes that the probabilistic distribution of a stochastic process is determined by its current state. Denote f as a l-fragment with $l \geq 3$ and $f[1:l-1]$ as its subfragment without the last location. We consider a stochastic process of $f[1:l-1]$ to append its next location, where the generation of f is equivalent to the event that the next location of $f[1:l-1]$ is $a_l^{<f>}$. We apply the $l-1$ order Markov independent assumption, indicating that the possibility of the next location is dependent on the previous $l-1$ locations (denote its possibility as $\mathbb{P}(a_l^{<f>}|f[1:l-1])$), and estimate the support count of f as follows.

$$\begin{aligned}\hat{\mathcal{S}}(f) &= \mathcal{S}(f[1:l-1]) \times \mathbb{P}(a_l^{<f>}|f[1:l-1]) \\ &\approx \mathcal{S}(f[1:l-1]) \times \mathbb{P}(a_l^{<f>}|f[2:l-1]) \\ &\approx \mathcal{S}(f[1:l-1]) \times \frac{\mathcal{S}(f[2:l])}{\mathcal{S}(f[2:l-1])}. \quad (13)\end{aligned}$$

Algorithm 1 Client response

Input: Local trajectory in the client: t; noise factor: η; a list of fragment candidates: $\mathcal{C}$; number of candidates: n.
Output: Response on the candidates: $\mathcal{R}$

```
function RESPOND($t, \eta, \mathcal{C}, n$)
    $\mathcal{R} \leftarrow$ an empty list of bits with length $n$
    for $i \leftarrow 1, ..., n$ do
        $b \leftarrow \mathbb{I}(\mathcal{C}[i]$ is a fragment of $t)$
        $b' \leftarrow$ randomize $b$ with (15)
        $\mathcal{R}[i] \leftarrow b'$
    end for
    return $\mathcal{R}$
end function
```

Then, for a l-fragment candidate set $\mathcal{C}$ with $l \geq 3$ from the Apriori scheme, we provide an initial estimation of the support counts. Since (13) works on the true support count, we first derive the expectation of the true support count of the shorter fragments by the inverse calculation of (5). Since (13) only provides coarse-grained estimations, we remove a candidate c only when its estimation $\hat{\mathcal{S}}(c)$ is far lower than k. Practically, we remove a candidate c when

$$\hat{\mathcal{S}}(c) < \lambda k, \quad (14)$$

where k is the parameter of (k, ξ)-anonymity and λ is a tunable parameter.

5.3 Client Response

In this part, we show how the server and clients collaborate to satisfy LDP on the client side and to provide high-quality responses about the local data.

Randomized response: Randomized response is a key component in our design, which is essential for the enforcement of LDP. Definition 5 provides randomization for binaries, which fits the form of clients' responses.

Definition 5. *(Randomized binary response) Given an input bit b, and noise factor $\eta \in [0, 0.5)$, randomized response outputs b', a noisy version of b, following the below rules.*

$$\mathbb{P}(b' = 0) = \begin{cases} 1-\eta, & \text{if } b = 0, \\ \eta, & \text{if } b = 1. \end{cases} \quad (15)$$

Client response: FASTPub enforces LDP by involving randomization of the clients' response. In this phase, the server first randomly selects the participating clients, and each client receives random candidates it responds to. To protect the clients from fingerprint tracing, a number of clients share the same set of candidates. Then, the client checks its local trajectory whether each candidate is present, and generates the true binary response for each candidate. After that, the client randomizes each of the true responses by the randomized binary response, and uploads the noisy version of the responses to the server. The procedure is present in Algorithm 1. Both the client selection and candidate selection should be random and unbiased, which is essential for the theoretical guarantees. Each participating client is recommended to participate in only one round of FASTPub, so that the privacy preservation level will not degrade due to the composition theorem of LDP.

Original trajectory: $a_1 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_2 \rightarrow a_1 \rightarrow a_1 \rightarrow a_4 \rightarrow a_3$

Candidate	$a_3 \rightarrow a_4 \rightarrow a_5$	$a_1 \rightarrow a_2 \rightarrow a_3$	$a_2 \rightarrow a_1 \rightarrow a_1$	$a_2 \rightarrow a_1 \rightarrow a_5$
Real response	True	False	True	False
Noisy response	*False*	False	True	False

Fig. 3. A client response example

An example of clients' responding to the candidates is given in Fig. 3. In this example, a client holds a trajectory with length 9 and receives four 3-fragments as candidates. It first derives the real response whether the candidate is a fragment of its local trajectory. It then randomizes this information to enforce LDP, and uploads the noisy response to the server.

5.4 Fragment Filtering

After aggregating the binary responses about the candidate, the server can only reserve a part of them with large support counts. The filtering procedure is with two goals, first, the remaining fragments should be capable to be the input of the Apriori-based candidate generation in the next round; second, the generated candidates in the next round should satisfy the anonymity loss restriction.

A natural consideration is to enforce k-anonymity on the uploaded responses centrally. It works in the traditional scenarios, but fails in our case because 1) the clients might not tell the truth, due to the randomized response, and 2) only a sampled portion of the clients participate in each round. In FASTPub, we require the server to filter the candidates and reserve those that satisfy k-anonymity with a limited error rate, or (k, ξ)-anonymity.

Definition 6 ((k, ξ)-anonymity)**.** *A data publication algorithm satisfies k-anonymity with an error rate ξ, or (k, ξ)-anonymity when*

$$\mathbb{P}(\mathcal{S}(d) \geq k) \geq 1 - \xi, \ \forall d \in \mathcal{P}, \qquad (16)$$

where $\mathcal{P}$ are all the published data of the algorithm.

In the candidate filtering phase, the server only reserves the candidates satisfying (k, ξ)-anonymity, which can be served as inputs of Apriori-based candidate generation, and have a theoretical bound of anonymity loss. When no candidate is admitted by the (k, ξ)-anonymity mechanism, it reveals that no new fragment can be published with the privacy requirement, and the system should be terminated without publishing any trajectory fragment.

(k, ξ)-anonymity provides a strong limit on the "precision" of candidate filtering, that every candidate accepted by (k, ξ)-anonymity is believed to satisfy k-anonymity with sufficient confidence. It bounds the risk of privacy leakage and helps the satisfaction of loss-bounded k-anonymity. However, the mechanism does not guarantee the "recall" of candidate filtering. Even if a candidate can be published by k-anonymity mechanism from the central perspective, it may fail to be published by FASTPub for any probability. As a result, FASTPub trade-offs some potential data utility for better privacy preservation.

The whole procedure of FASTPub is demonstrated in Algorithm 2. Note that the derivation of noise factor (η_l) and support count threshold for (k, ξ)-anonymity (S_l) will be provided and theoretically proven in the next part.

Algorithm 2 FASTPub: Overall structure

```
Input: Trajectory data in the clients: $t$; the set of clients: $\mathcal{N}$;
    the number of clients: $N$; the portion of participating
    clients in each round: $M$; target fragment length: $l_{max}$
Output: Published trajectory fragments
for $l \leftarrow 1, ..., l_{max}$ do
    if $l = 1$ then
        $\mathcal{C}_l \leftarrow$ all possible 1-fragments
    else
        Generate $\mathcal{C}_l$ based on $\mathcal{P}_{l-1}$ with (12)
    end if
    if $l \geq 3$ then
        Remove $c \in \mathcal{C}_l$ when (14) holds
    end if
    if $\mathcal{C}_l$ is empty then
        return with no result
    end if
    $\eta_l \leftarrow$ noise factor for $l$-fragments
    $S_l \leftarrow$ support count threshold for $l$-fragments
    $D \leftarrow$ draw $M \cdot N$ clients from $\mathcal{N}$ (not drawn before)
    Initialize support counts for $\mathcal{C}_l$
    for $d \in D$ do
        $\mathcal{C}_l^{<d>} \leftarrow$ sample candidates form $\mathcal{C}_l$
        $\mathcal{R} \leftarrow$RESPOND$(t_d, \eta_l, \mathcal{C}_l^{<d>}, |\mathcal{C}_l^{<d>}|)$
        Update support counts with $\mathcal{R}$
    end for
    $\mathcal{P}_l \leftarrow \{f \in \mathcal{C}_l \mid \mathcal{S}(f) \geq S_l\}$
end for
return $\mathcal{P}_{l_{max}}$
```

5.5 Criteria Analysis

In this part, we prove that FASTPub satisfies loss-bounded k-anonymity and LDP. In addition, we fulfill the incomplete parts of Algorithm 2, that the derivation of η_l and S_l.

We first verify the satisfaction of LDP. Based on Definition 2, we conclude the following theorem.

Theorem 2. *(LDP) The client side scheme of FASTPub algorithm meets $\left(n \ln \dfrac{1-\eta}{\eta}\right)$-LDP, where n indicates the number of candidates sent to the client.*

Proof. See Appendix A. □

Theorem 2 enables us to determine the noise factor η_l in each round after fixing ϵ. In particular, we notice that if the number of candidates sent to the clients n is too large, the noise factor will concomitantly increase, which decreases the data utility. In practice, we can relieve the harm by setting a limit on the number of candidates sent to each client (C): if the number of candidates exceeds C in some iterations, each client will only receive and respond to C candidates, instead of all candidates.

For the satisfaction of (k, ξ)-anonymity, we need to determine the threshold of support count, which will be used to check whether a candidate can be admitted by the server to satisfy (k, ξ)-anonymity. The Hoeffding inequality is a powerful tool in the estimation of randomized response and sampling [40]. Based on the Hoeffding inequality, we take both randomized response and sampling into consideration and derive the support count threshold being concluded into the following theorem.

Theorem 3. *(Candidate filtering threshold) Any l-fragment f satisfies (k, ξ)-anonymity if its support count exceeds S_l, which is given by*

$$S_l = M\left(\frac{k}{N}(1-\eta) + \frac{N-k}{N}\eta + \sqrt{\frac{-\ln \xi}{2M}}\right), \tag{17}$$

where N is the number of all clients, M is the number of participating clients giving a response on f, and η is the noise factor in the corresponding round.

Proof. See Appendix B. □

Finally, we consider the satisfaction of loss-bounded k-anonymity, which provides local privacy preservation for clients before any uploads. Since the FASTPub is a symbiosis with randomization, we cannot firmly bound the anonymity loss, so we bound its expectation as the following theorem shows.

Theorem 4. *(Loss-bounded k-anonymity) For any l-fragment f published by a client, its expectation of anonymity loss is bounded by*

$$\mathbb{E}\big(\tau(f)\big) \leq \sum_{x=1}^{l} \xi^{\frac{x^2+x-2}{2}} \tag{18}$$

where ξ is the allowed error rate of (k, ξ)-anonymity.

Proof. See Appendix C. □

In practice, when the value ξ is low (*e.g.* 0.01 in our implementation), the increase of value in (18) becomes neglectable when $x > 1$, which means that the expectation of any l-fragment will be closed to 1. Consequently, for any client publishing an l-fragment f, it has high confidence that the publication will at most expose one sensitive location in the fragment in the head or the tail. The overall anonymity loss bound for a client can be derived by simply multiplying the value in (18) and the number of the received candidate.

6 Evaluations

6.1 Experiment setup

In this section, we evaluate the data utility of the trajectory database generated by FASTPub. The data utility of a trajectory database is usually evaluated based on the performance of downstream tasks. In this paper, we choose two representative trajectory data mining tasks: *frequent sequence mining* and *count query* [10], [11]. Frequent sequence mining finds out trajectory sequences happen in the population with a frequency higher than a threshold. Count query returns the count or frequency of a given sequence.

Dataset: The experiments are performed on two datasets: *MSNBC* and *Oldenburg*. MSNBC dataset [41] stores trajectories of user's browsing history on msnbc.com by category, and Oldenburg dataset is created by the Brinkhoff's data generator [42] and contains synthetically generated trajectories of objects traveling in the city of Oldenburg. In order to simulate the real-world large-scale trajectory publication scenario, we filter out the uninformative records (trajectories with lengths smaller than 3) in MSNBC and duplicate the remainings 100 times. For Oldenburg, we convert locations in trajectories into distinct locations of interest via grid matching. Key properties of the datasets are summarized in Table 2. Due to the space limit, the results of Oldenburg dataset are presented in Appendix D.

TABLE 2
Summary of the datasets

	MSNBC	*Oldenburg*
Num. location	17	64
Num. trajectory	4.7×10^7	2.0×10^7
Avg. trajectory length	8.55	7.96

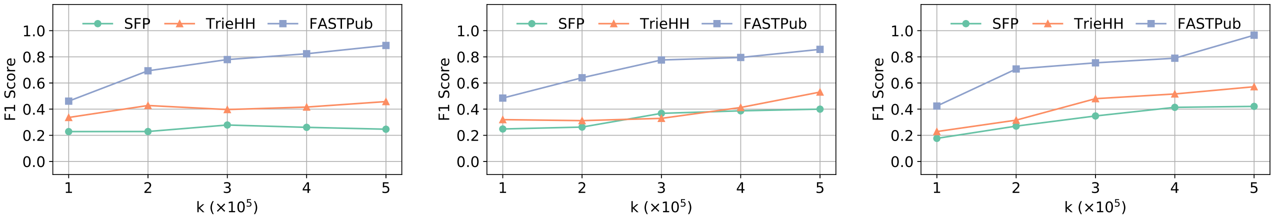
Benchmark: We use two benchmarks to test the efficiency of our mechanism: Apple's SFP [23] and Google's TrieHH [19]. SFP is an industrial solution of LDP available in trajectory publication, which tackles the challenge of high dimensionality by building a frequency oracle to decrease the number of possible fragments to be checked. TrieHH is designed for discovering popular strings (heavy hitters) in edge data. It can be adapted for trajectory data with little modification because strings have almost the same structure as trajectories. It is another representative work under the umbrella of FA. However, it trade-offs the local privacy with system utility, resulting in lower privacy preservation than FASTPub in the untrusted environment. We compare FASTPub with this utility-driven design, and show that FASTPub can achieve both higher utility and stricter privacy.

Parameter: The k values are set to be between 100,000 and 500,000, equivalently identifying minimum frequencies between 0.21% and 2.5% in the whole dataset. In each round, the portion of participating clients M is set to 0.2, so that each client is restricted to participating only once. The default value of LDP parameter ϵ is set to 10.0, which is within the range recommended by [21]. The threshold of candidate cleaning λ is set to 0.8. The allowed error rate ξ is set to 0.01. The default number of candidates sent to each client is set to $C = 5$. For TrieHH, in order to guarantee fairness regarding client usage, we restrict the maximum number of rounds to be consistent with FASTPub. The design of TrieHH makes ϵ and the number of participants depends on each other, so we do not require TrieHH to provide the same privacy preservation level as other algorithms. The other parameters of SFP and TrieHH are consistent with their original implementations [19], [23].

6.2 Frequent sequence mining

Frequent sequence mining is a practically important data mining task on sequential (trajectory) databases. It outputs a list of patterns (continuous subsequence) that presents in the population with a frequency higher than a given threshold. The performance of frequent sequence mining is usually measured by F1 score, which comprehensively considers the precision and recall to find the correct frequent sequences.[6] Notably, the mechanism of FASTPub naturally includes the procedure of filtering out infrequent subsequences, indicating that every fragment published by FASTPub is believed to have a minimum frequency decided by the k-anonymity

[6] F1 score equals $\frac{2pr}{p+r}$, where p is precision and r is recall.

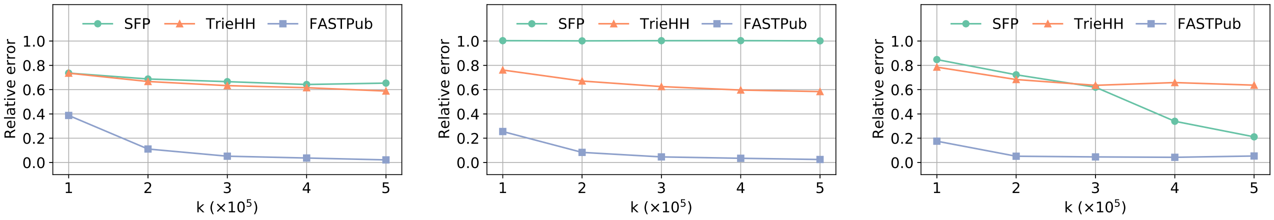


(a) 3-length sequences

(b) 4-length sequences

(c) 5-length sequences

Fig. 4. Performance of different methods in frequent sequence mining task, with different values of target sequence length and k.



(a) 3-length query

(b) 4-length query

(c) 5-length query

Fig. 5. Performance of different methods in count query task, with different values of query length and k.

setting. Therefore, for the frequent sequence mining tasks, we set the target frequency to be consistent with the k value (target frequency $f = k/N$). It reveals the maximal capacity of FASTPub database in frequent sequence mining. The frequent mining tasks aim at mining sequences with non-trivial lengths $3 \sim 5$.[7]

The performance (F1 score) of FASTPub and the benchmarks are shown in Fig. 4. In our settings, FASTPub not only has better performances than SFP, but also outperforms TrieHH, which provides weaker privacy preservation. Numerically, among the 15 settings, FASTPub gains an average precision, recall, and F1 score of 1.00, 0.59, and 0.72 respectively. Compared to the two benchmarks, it gains a higher average F1 score for $1.8 \sim 2.4$ times. The high performance is owing to the ingenious utilization of the property of trajectory data, which significantly reduces the utility loss introduced by randomization. On the contrary, SFP gains a poor data utility for two reasons. First, it forces the clients to trim the original trajectory to a fixed-length fragment, which leads to significant information loss for a long trajectory. Second, it uploads large count mean sketches (1024 bits in our setting), and requires significant noise to satisfy LDP. TrieHH, which provides weaker privacy preservation, turns out to be outperformed by FASTPub, because the clients can only upload fragments starting at the first element, which leads to information loss.

6.3 Count query

Count query task is another important trajectory data mining task that aims at returning the frequency of any given query sequence. The performance of the count query is measured by the relative error given as follows.

$$error(Q(\tilde{D})) = \frac{|Q(\tilde{D}) - Q(D)|}{Q(D)}, \qquad (19)$$

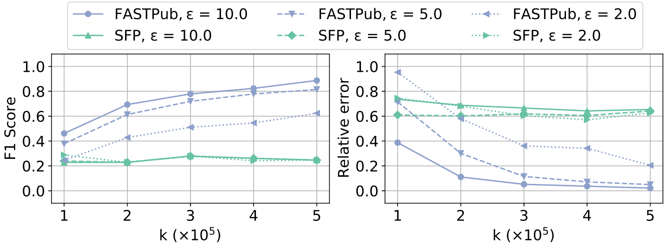
where $Q(\tilde{D})$ and $Q(D)$ denotes the outputs from the evaluated database and the original true database, respectively.

Notably, returning the query results of infrequent sequences naturally breaks the privacy constraint of k-anonymity. Therefore, we denote the queries on fragments with a true count higher than k as legal queries, and calculate the median relative error of all legal queries with a particular length as the measurement of data utility. SFP is able to answer any query on fix-length fragments, but FASTPub and TrieHH may mistakenly filter out the results of some legal queries. In such cases, we use the Markov independent assumption-based support count estimation in (13) to derive the query response.

The performance (relative error) of FASTPub and benchmarks are shown in Fig. 5 covering different k values and query length between $3 \sim 5$. The maximal fragment length l_{max} is set to be equal to the query length in order to reveal the maximal potential of FASTPub. In all the settings, FASTPub outperforms TrieHH and SFP by providing a more precise count query with a smaller relative error. Among 15 settings, FASTPub, TrieHH, and SFP gain average relative errors of 0.09, 0.66, 0.74, respectively. As a result, the privatization of FASTPub only introduces errors of $12.8\% \sim 14.4\%$ compared to the benchmarks. The reasons are consistent with those in Section 6.2. Notably, SFP performs fairly badly in 4-length queries due to the trimming strategy of SFP. SFP requires the client to trim the raw trajectory into a predefined even length. odd-length queries can achieve relatively good performance when the trimming length is one more than the query length. On the contrary, SFP has to set the trimming length far higher than or equal to the query length for even-length queries, which leads to a significant degradation caused by trimming loss and query padding.[8]

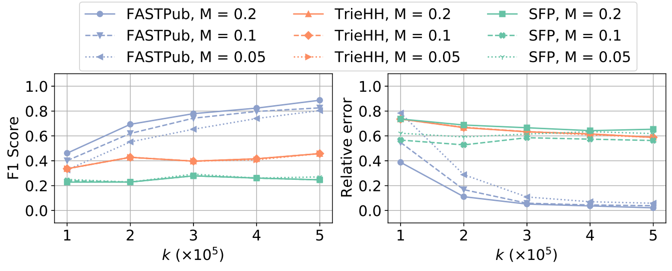
[7] Sequences with length ≤ 2 are too trivial as mining results, and there are too few frequent sequences as ground truth with length ≥ 6.

[8] We experiment with different trimming lengths for SFP, and all results in this paper are the optimal performance among our experiments.



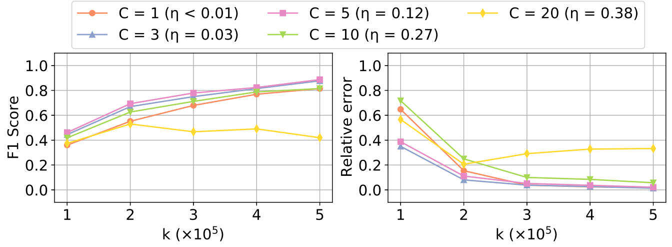
(a) Frequent sequence mining (b) Count query

Fig. 6. Performance of LDP algorithms under different ϵ.



(a) Frequent sequence mining (b) Count query

Fig. 7. Performance of three algorithms under different M.



(a) Frequent sequence mining (b) Count query

Fig. 8. Performance of FASTPub under different C.

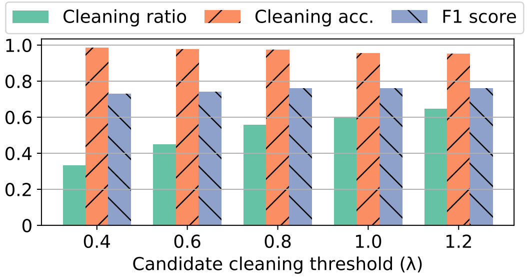


Fig. 9. Performance of FASTPub under different λ.

6.4 Sensitivity analysis

Privacy preservation level: The strength of privacy preservation from LDP is controlled by ϵ, where lower ϵ indicates stronger privacy preservation. For LDP algorithms like FASTPub or SFP, a lower ϵ forces the clients' responses to be more randomized. To investigate its effect, we run FASTPub and SFP to publish 3-fragments with ϵ to be 5.0 or 2.0, and compare them with the default settings where $\epsilon = 10.0$. The results are shown in Fig. 6. First, FASTPub can still outperform SFP in most of the cases with lower ϵ. In addition, FASTPub has worse performance when we use a lower ϵ, because more noise has to be added to the response when ϵ gets lower. Especially, when $\epsilon = 2.0$, FASTPub performs worse than SFP in some particular settings. On the other hand, SFP is not that sensitive for ϵ. We conclude the reason for the insensitivity of SFP as follows: even if ϵ is high, SFP is only able to find out several fragments, whose frequency is much higher than k and other fragments. Therefore, the results are hardly influenced by adding more noise. Meanwhile, FASTPub strives to publish all fragments satisfying the requirement, including the fragments whose frequency is slightly higher than k, which is highly sensitive to the amount of noise added.

Number of participants: The effectiveness of LDP algorithms requires the number of participants to be large [21]. To evaluate such dependency, we run FASTPub, TrieHH, and SFP to publish 3-fragments with the portion of participants per round M to be 0.1 and 0.05, and compare the results with the default portion, 0.2. The results in Fig. 7 show that the performance of FASTPub is dependent on a large number of participants. The reason is that FASTPub needs a large number of participants to inhibit the randomness, so that the true support count can be reliably estimated. On the other hand, TrieHH and SFP are not sensitive to M. The reason for the insensitivity of SFP is similar to the above paragraph. Compared to FASTPub, whose efficiency is suffered from both randomization and sampling, TrieHH, which does not apply randomization, is only influenced by sampling, whose degrading effect has been low enough when $M = 0.05$.

More response or cleaner response? Parameter C controls the number of candidates that each client should respond to. If we require each client to respond to all of them, according to Theorem 2, the randomization factor η will be closed to 0.5, leading to the poor utility of response. To tackle the problem, we set a parameter C, which controls the maximum number of candidates should each client respond. There exists a trade-off regarding the choice of C: setting a high C will provide each candidate with more responses from the clients, but the quality of the responses will be decreased due to high η. Meanwhile, setting a low C guarantees the truthfulness of the responses, but each candidate can only be checked by a small number of clients.

To evaluate the effect of choosing different C, we record the results of FASTPub to publish 3-fragments under different settings of C and the corresponding η. The results are shown in Fig. 8. For two tasks, we can see that $C = 5$ has the best performance in balancing the trade-off between quantity and quality. In addition, the performance of FASTPub is not significantly sensitive to the settings of C, since a similar performance is obtained when C is set to 3. In particular, an anomaly occurs in the Oldenburg dataset when $C = 20$. The reason is consistent with that in Fig. 10(c), that the number of remaining candidates becomes smaller than C, which decreases the noise factor η, and increases the quality of client responses.

Candidate cleaning threshold: The candidate cleaning threshold λ controls the strictness of Markov independent assumption-based candidate cleaning. When a larger λ is

TABLE 3
Summary of the communication overheads. The values represent expected size of the transmitted data structure of each client.

	Download (bit)	Upload (bit)
SFP	0	4096
TrieHH	1680	64
FASTPub	400	5

applied, FASTPub removes more candidates in the cleaning phase. It increases the risk of mistakes in removing frequent candidates, but also increases the number of responses of the remaining candidates. To investigate the effect of the candidate cleaning threshold, we conduct experiments with different λ values. The experiments are conducted on frequent pattern mining task, with $l_{max} = 4$ and $k = 300,000$. We focus on three metrics. 1) cleaning ratio: the portion of removed candidates in the candidate cleaning phase. 2) cleaning accuracy: the portion of correctly removed candidates (whose ground truth frequency is smaller than k) in the removed candidates. 3) F1 score: performance of frequent pattern mining task.

The experiment results are shown in Fig. 9, where the λ values are set to 0.4, 0.6, 0.8, 1.0, and 1.2. The cleaning ratio increases progressively with respect to λ, indicating a stricter candidate cleaning. It also increases the portion of mistakenly cleaned candidates, although the portion is always smaller than 5% even for $\lambda = 0.2$. Lastly, the F1 score has a slight increase when we use stricter candidate cleaning, revealing its effectiveness in improving data utility. The data utility of FASTPub is generally not sensitive with λ, and the application of candidate cleaning does not require fine-tuning of λ.

6.5 Communication overhead

The communication overhead is critical in evaluating trajectory publication algorithms. We measure the average transmitted bits of the algorithms, both downloads and uploads, and summarize the results in Table 3.[9] We only consider the average size of data structures required for transmission by the algorithms, and ignore the overheads introduced by the transmission protocol. SFP, which is not interactive, does not require any downloaded data, but requires uploading two count mean sketches with huge sizes. TrieHH requires the clients to download a large prefix tree, and upload an added leaf on the tree. It results in a large download size and a small upload size. FASTPub gains the best performance in total communication size. It requires the clients to download several candidate fragments, and to upload one bit response to each candidate. Therefore, the communication overhead of FASTPub can be further reduced by sending fewer candidates to each client with a lower C.

7 Conclusions

The growing awareness of data privacy calls for mining global knowledge with no transmission of raw data from end devices. In this paper, we proposed a privacy-preserving trajectory publication mechanism, named as FASTPub, to facilitate further data analytic tasks in such untrusted environments. Following the emerging FA paradigm, we transform the one-shot trajectory data collection scheme into an interactive trajectory construction scheme between the server and clients. Our mechanism enforces both loss-bounded k-anonymity and LDP. The utility of our mechanism is greatly improved by introducing fragmentation and exploiting the Apriori property and Markov independent assumption. Experimental results show that, FASTPub gains better performances in two representative downstream tasks ($2.5 \sim 6.1$ times higher F1 score and $85.6\% \sim 87.2\%$ less relative error) than an industrial LDP method and another weak-privacy-guaranteed FA based mechanism.

[9] We encode each location data as an integer (32 bits).

References

[1] 2017 GNSS Market Report. European global navigation satellite system agency. [Online]. Available: https://www.gsa.europa.eu/2017-gnss-market-report

[2] H. Wang, S. Zeng, Y. Li, and D. Jin, "Predictability and prediction of human mobility based on application-collected location data," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2457–2472, Jul. 2021.

[3] C. Guo, B. Yang, J. Hu, and C. Jensen, "Learning to route with sparse trajectory sets," in *IEEE Int. Conf. Data Eng.*, Paris, France, Apr. 2018, pp. 1073–1084.

[4] Y. Zhu, Y. Wu, and B. Li, "Trajectory improves data delivery in urban vehicular networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 1089–1100, Apr. 2014.

[5] F. Basık, B. Gedik, Ç. Etemoğlu, and H. Ferhatosmanoğlu, "Spatio-temporal linkage over location-enhanced services," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 447–460, Feb. 2017.

[6] Z. Peng, Y. Yao, B. Xiao, S. Guo, and Y. Yang, "When urban safety index inference meets location-based data," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2701–2713, Nov. 2018.

[7] Y. Zheng, "Trajectory data mining: an overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, May 2015.

[8] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen *et al.*, "The mobile data challenge: Big data for mobile computing research," Tech. Rep., 2012.

[9] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, San Diago, CA, Aug. 2011, pp. 316–324.

[10] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos, "Local suppression and splitting techniques for privacy preserving publication of trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1466–1479, Jul. 2017.

[11] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proc. ACM Conf. Comput. Commun. Secur.*, Raleigh, NC, Oct. 2012, pp. 638–649.

[12] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 264–278, Mar. 2018.

[13] N. Mohammed, B. C. Fung, and M. Debbabi, "Walking in the crowd: anonymizing trajectory data for pattern analysis," in *Proc. ACM Conf. Inf. knowl. Manage.*, Hong Kong, China, Nov. 2009, pp. 1441–1444.

[14] 2018 reform of EU data protection rules. European Commission. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf

[15] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.*, vol. 10, no. 05, pp. 571–588, Jul. 2002.

[16] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proc. ACM -SIGACT-SIGART Symp. Princ. Database Syst.*, San Diego, CA, Jun. 2003, pp. 211–222.

[17] P. Kairouz, B. McMahan, and V. Smith. Federated learning and analytics: Industry meets academia. NeurIPS 2020 tutorial. [Online]. Available: https://sites.google.com/view/fl-tutorial/home

[18] Federated analytics: Collaborative data science without data collection. Google AI. [Online]. Available: https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html

[19] W. Zhu, P. Kairouz, B. McMahan, H. Sun, and W. Li, "Federated heavy hitters discovery with differential privacy," in *Proc. Int. Conf. Artif. Intell. Statist.*, Palermo, Italy, Jun. 2020, pp. 3837–3847.

[20] Z. Wang, Y. Zhu, D. Wang, and Z. Han, "Fedacs: Federated skewness analytics in heterogeneous decentralized data environments," in *Proc. IEEE/ACM Int. Symp. Qual. Service*, virtual meeting, Jun. 2021, pp. 1–10.

[21] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM Conf. Comput. Commun. Secur.*, Scottsdale, AZ, Nov. 2014, pp. 1054–1067.

[22] M. E. Gursoy, L. Liu, S. Truex, and L. Yu, "Differentially private and utility preserving publication of trajectory data," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2315–2329, Oct. 2018.

[23] Apple, "Learning with privacy at scale," *Apple Machine Learning Journal*, vol. 1, no. 8, 2017.

[24] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, and W. Yang, "Local differential private data aggregation for discrete distribution estimation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2046–2059, Sep. 2019.

[25] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, Mar. 2020.

[26] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity," in *IFIP Annu. Conf. Data and Appl. Secur. Privacy*, Storrs, CT, Aug. 2005, pp. 166–177.

[27] G. Zhong and U. Hengartner, "A distributed k-anonymity protocol for location privacy," in *Int. Conf. Pervasive Comput. Commun.*, Galveston, TX, Mar. 2009, pp. 1–10.

[28] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preservation in location-based services: a novel metric and attack model," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3006–3019, Oct. 2021.

[29] P. Huang, X. Zhang, L. Guo, and M. Li, "Incentivizing crowdsensing-based noise monitoring with differentially-private locations," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 519–532, Feb. 2021.

[30] J. Liu, C. Zhang, B. Lorenzo, and Y. Fang, "Dpavatar: A real-time location protection framework for incumbent users in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 552–565, Mar. 2019.

[31] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv*, 2019.

[32] G. Cormode and I. L. Markov, "Bit-efficient numerical aggregation and stronger privacy for trust in federated analytics," *arXiv*, 2021.

[33] G. Cormode, T. Kulkarni, and D. Srivastava, "Marginal release under local differential privacy," in *Proc. ACM Int. Conf. Manage. Data*, Houston, TX, Jun. 2018, pp. 131–146.

[34] D. Ravichandran and S. Vassilvitskii, "Evaluation of cohort algorithms for the FLoC API." [Online]. Available: https://github.com/google/ads-privacy/raw/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf

[35] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Improving the utility of differentially private data releases via k-anonymity," in *IEEE Int. Conf. Trust Secur. Privacy Comput. Commun.*, Melbourne, Australia, Jul. 2013, pp. 372–379.

[36] N. Li, W. H. Qardaji, and D. Su, "Provably private data anonymization: Or, k-anonymity meets differential privacy," *CoRR*, vol. 49, p. 55, 2011.

[37] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Enhancing data utility in differential privacy via microaggregation-based k-anonymity," *VLDB J.*, vol. 23, no. 5, pp. 771–794, Feb. 2014.

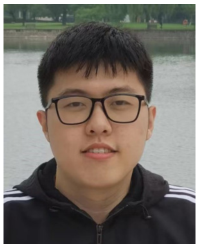
[38] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory Cryptography Conf.* New York: Springer, Mar. 2006, pp. 265–284.

[39] R. Agarwal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. VLDB Conf.*, vol. 487, Santiago de Chile, Chile, Sep. 1994, p. 499.

[40] D. Wang and Z. Han, *Sublinear algorithms for big data applications*. Springer, 2015.

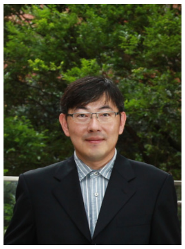
[41] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of navigation patterns on a web site using model-based clustering," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, Boston, MA, Aug. 2000, pp. 280–284.

[42] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, Jun. 2002.



Zibo Wang received the B.E. degree in Electrical and Computer Engineering from Shanghai Jiao Tong University in 2020. He is currently pursuing the Ph.D. degree in the same institute. His research interests include federated learning, federated analytics, and privacy computing.

Yifei Zhu (Member, IEEE) is currently an Assistant Professor at Shanghai Jiao Tong University, China. He received the B.E. degree from Xi'an Jiaotong University, China, in 2012, and the M.Phil. degree from The Hong Kong University of Science and Technology, China, in 2015, and the Ph.D degree in Computer Science from the Simon Fraser University, Canada, in 2020. His current research interests include edge computing, multimedia networking, distributed machine learning systems, where he published in ACM SIGCOMM, IEEE INFOCOM, ACM Multimedia, and many other venues.



Dan Wang (Senior Member, IEEE) receives his B.Sc., M.Sc., Ph.D. from Peking University, Case Western Reserve University and Simon Fraser University, all in Computer Science. His research falls in general computer networking and systems, where he published in ACM SIGCOMM, ACM SIGMETRICS and IEEE INFOCOM, and many others. He is the steering committee chair of IEEE/ACM IWQoS. He served as the TPC co-Chair of IEEE/ACM IWQoS 2020. His recent research focus on smart energy systems. He won the Best Paper Awards of ACM e-Energy 2018 and ACM Buildsys 2018. He has served as a TPC co-Chair of the ACM e-Energy 2020 and he will serve as General co-Chair of the ACM e-Energy 2022. He is a steering committee member of ACM e-Energy. He serves as a founding area editor of ACM SIGEnergy Energy Informatics Review. His research has been adopted by industry, e.g., Henderson, Huawei, and IBM. He won the Global Innovation Award, TechConnect, in 2017.

Zhu Han (Fellow, IEEE) received his M.S. and Ph.D. from the University of Maryland in 1999 and 2003, respectively. He is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless networking, game theory, big data analysis, security, and smart grid.

Appendix A
Proof of Theorem 2

For each publication, the clients generate real responses $\mathcal{R}$ on candidates $\mathcal{C}$, which is in the form of binary and has length n, and generate the corresponding noisy version $\mathcal{R}'$. We consider any desired output Y, which is in the same form as $\mathcal{R}$ or $\mathcal{R}'$. For each bit of Y, $\mathcal{R}$, and $\mathcal{R}'$ (denoted the index as i), the possibility of $Y[i] = \mathcal{R}'[i]$ is given by the randomized binary response (15):

$$\mathbb{P}(\mathcal{R}'[i] = Y[i]) = \begin{cases} 1-\eta, & \text{if } Y[i] = \mathcal{R}[i], \\ \eta, & \text{if } Y[i] \neq \mathcal{R}[i]. \end{cases} \tag{20}$$

Since the randomization of each bit is independent, the possibility of $Y = \mathcal{R}'$ is given by

$$\mathbb{P}(\mathcal{R}' = Y) = \prod_{i=1,..,n} \mathbb{P}(\mathcal{R}'[i] = Y[i]) \tag{21}$$

When considering the satisfaction of LDP, we only need the minimum and maximum of $\mathbb{P}(\mathcal{R}' = Y)$. Obviously, it reaches the maximum when the possibility for bits in (21) all take their maximum, $1-\eta$, in (20), and reaches the minimum when all bits take their minimum η, *i.e.*,

$$\max[\mathbb{P}(\mathcal{R}' = Y)] = \prod_{i=1,..,n} (1-\eta) = (1-\eta)^n, \tag{22}$$

$$\min[\mathbb{P}(\mathcal{R}' = Y)] = \prod_{i=1,..,n} \eta = \eta^n. \tag{23}$$

Then the satisfaction of LDP yields

$$e^{\epsilon} \geq \max\left[\frac{\mathbb{P}(\mathcal{R}'_1 = Y))}{\mathbb{P}(\mathcal{R}'_2 = Y))}\right] = \frac{(1-\eta)^n}{\eta^n}, \tag{24}$$

which yields

$$\epsilon \geq n \ln \frac{1-\eta}{\eta}. \tag{25}$$

We consider the tightest satisfaction of LDP and conclude Theorem 2.

Appendix B
Proof of Theorem 3

We consider a client holding trajectory t responds on a candidate fragment f. We denote $y = 1$ as the client respond "yes" about f, and $y = 0$ otherwise. Definition 5 provides the corresponding possibility values:

$$\mathbb{P}(y = 1) = \begin{cases} 1-\eta, & \text{if } f \text{ is a fragment of } t, \\ \eta, & \text{otherwise.} \end{cases} \tag{26}$$

Remind that there are N clients at all. Considering that there are k_0 clients holding fragment f as ground truth, we have the expectation of y can be given:

$$\mathbb{E}(y) = \frac{k_0}{N}(1-\eta) + \frac{N-k_0}{N}\eta. \tag{27}$$

The nature of k-anonymity is exactly $k_0 \geq k$. Since the value in (27) increases as k_0 increases, we can rewrite it to represent the restriction on $\mathbb{E}(y)$ when k-anonymity is satisfied:

$$k-\text{anonymity} \iff \mathbb{E}(y) \geq \frac{k}{N}(1-\eta) + \frac{N-k}{N}\eta. \tag{28}$$

We denote the bound of $\mathbb{E}(y)$ given in (28) as $\hat{y}$:

$$\hat{y} = \frac{k}{N}(1-\eta) + \frac{N-k}{N}\eta. \tag{29}$$

On the other hand, denote the support count of f (number of clients uploading f to the server) as $S(f)$, and the average value of y is given by

$$\overline{y} = \frac{S(f)}{M}, \tag{30}$$

where M is the number of clients responding on f.

The Hoeffding inequality is a powerful tool for estimating the divergence between the average of random variables and its expectation, which can be efficiently adapted to the estimation of $\overline{y}$ and $\mathbb{E}(y)$ as

$$\mathbb{P}\big(\overline{y} - \mathbb{E}(y) \geq \delta\big) \leq \exp(-2\delta^2 M). \tag{31}$$

Rewrite it,

$$\mathbb{P}\big(\mathbb{E}(y) \leq \overline{y} - \delta\big) \leq \exp(-2\delta^2 M). \tag{32}$$

For the satisfaction of (k,ξ)-anonymity, we hope the RHS of (32) is replaced by ξ. Since δ in (32) can be arbitrary value, we use

$$\delta = \sqrt{\frac{-\ln \xi}{2M}}, \tag{33}$$

so that (32) can be rewritten as

$$\mathbb{P}\Bigg(\mathbb{E}(y) \leq \overline{y} - \sqrt{\frac{-\ln \xi}{2M}}\Bigg) \leq \xi. \tag{34}$$

To bridge (34) with the satisfaction of (k,ξ)-anonymity, we start with an assumption $\mathcal{A}$

$$\mathcal{A}:\ \overline{y} - \sqrt{\frac{-\ln \xi}{2M}} \geq \hat{y}. \tag{35}$$

Supposed that $\mathcal{A}$ is satisfied, we have

$$\mathbb{P}\big(\mathbb{E}(y) \leq \hat{y}\big) \leq \mathbb{P}\Bigg(\mathbb{E}(y) \leq \overline{y} - \sqrt{\frac{-\ln \xi}{2M}}\Bigg) \leq \xi. \tag{36}$$

Eq. (28) shows that the satisfaction of k-anonymity is equivalent to $\mathbb{E}(y) \geq \hat{y}$, and therefore, (36) indicates the the possibility that k-anonymity fails is smaller than ξ, which is equivalent to (k,ξ)-anonymity. Therefore, the enforcement of $\mathcal{A}$ is sufficient for the satisfaction of (k,ξ)-anonymity.

By merging (29), (30) and (35), we can form the support count threshold for (k,ξ)-anonymity:

$$\overline{y} \geq \frac{k}{N}(1-\eta) + \frac{N-k}{N}\eta + \sqrt{\frac{-\ln \xi}{2M}}, \tag{37}$$

$$S(f) \geq M\Bigg(\frac{k}{N}(1-\eta) + \frac{N-k}{N}\eta + \sqrt{\frac{-\ln \xi}{2M}}\Bigg), \tag{38}$$

which yields Theorem 3.

Appendix C
Proof of Theorem 4

In FASTPub, all clients simply provide binary responses to the candidate fragments. Therefore, in this part, we prove the bound of anonymity loss for all candidate fragments. For any candidate l-fragment f, according to the definition of anonymity loss in (6), the maximum of $\tau(f)$ is l. To calculate $\mathbb{E}(\tau(f))$, we need to derive the possibility of $\tau(f) = 1, ..., l$ and sum them up.

We first consider the possibility of $\tau(f) \geq 1$, since there is no evidence for inferring about that, we can only derive a trivial bound as follows.

$$\mathbb{P}\big(\tau(f) \geq 1\big) \leq 1. \tag{39}$$

The derivation of larger $\tau(f)$ is based on the definition of (k, ξ)-anonymity and the Apriori property. For any trajectory data uploaded by a client, since it is served as a candidate inferred by the server, all fragments of it must satisfy k-anonymity (with no anonymity loss) with the possibility of $1 - \xi$.

Then we considers $\tau(f) \geq x > 1$. It implies that all fragments with a length between $l - x + 1$ and l fail to meet k-anonymity. Therefore, denote the set of fragments of f with length between $l - x + 1$ and l as $\mathcal{F}_x(f)$, the possibility of $\tau(f) >= x$ is equivalent to the possibility that all fragments in $\mathcal{F}_x(f)$ (including f itself) fail to satisfy k-anonymity synchronously. Since the failure of different fragments is independent due to the procedure of FASTPub, we can calculate the possibility of synchronous failures by simply multiplying them, *i.e.*,

$$\begin{aligned}\mathbb{P}\big(\tau(f) \geq x\big) &= \prod_{\forall f' \in \mathcal{F}_x(f)} \mathbb{P}(\mathcal{S}(f') < k) \\ &\leq \prod_{\forall f' \in \mathcal{F}_x(f) \setminus f} \mathbb{P}(\mathcal{S}(f') < k) \\ &= \prod_{\forall f' \in \mathcal{F}_x(f) \setminus f} \xi. \end{aligned} \tag{40}$$

To derive the value in (40), we need to calculate the number of fragment in $\mathcal{F}_x(f) \setminus f$. Obviously, for a target fragment f with length l, the number of $(l-x)$-fragments ($0 \leq x \leq l-1$) of f is equal to $x+1$. Therefore, the number of fragments in $\mathcal{F}_x(f) \setminus f$ can be expressed as

$$\begin{aligned}|\mathcal{F}_x(f) \setminus f| &= \sum_{i=l-x+1}^{l-1} l - i + 1 \\ &= \frac{x^2 + x - 2}{2}. \end{aligned} \tag{41}$$

Therefore,

$$\mathbb{P}\big(\tau(f) \geq x\big) \leq \xi^{\frac{x^2+x-2}{2}}. \tag{42}$$

Specially, when $x > l$, we have

$$\mathbb{P}\big(\tau(f) \geq x\big) = 0. \tag{43}$$

Obviously, we have

$$\mathbb{P}\big(\tau(f) = x\big) = \mathbb{P}\big(\tau(f) \geq x\big) - \mathbb{P}\big(\tau(f) \geq x + 1\big). \tag{44}$$

Note that (39) is also included by (42), so (42) and (44) hold universally for all $1 \leq x \leq l$.

Based on the derived equations in (42),(43), and (44), we can bound $\mathbb{E}(\tau(f))$ for any published l-fragment f:

$$\begin{aligned}\mathbb{E}\big(\tau(f)\big) &= \sum_{x=1}^{l} x\mathbb{P}\big(\tau(f) = x\big) \\ &= \sum_{x=1}^{l} x\Big(\mathbb{P}\big(\tau(f) \geq x\big) - \mathbb{P}\big(\tau(f) \geq x + 1\big)\Big) \\ &= \sum_{x=1}^{l} \mathbb{P}\big(\tau(f) \geq x\big) - l\mathbb{P}\big(\tau(f) \geq l + 1\big) \\ &\leq \sum_{x=1}^{l} \xi^{\frac{x^2+x-2}{2}} - 0 \\ &= \sum_{x=1}^{l} \xi^{\frac{x^2+x-2}{2}}, \end{aligned} \tag{45}$$

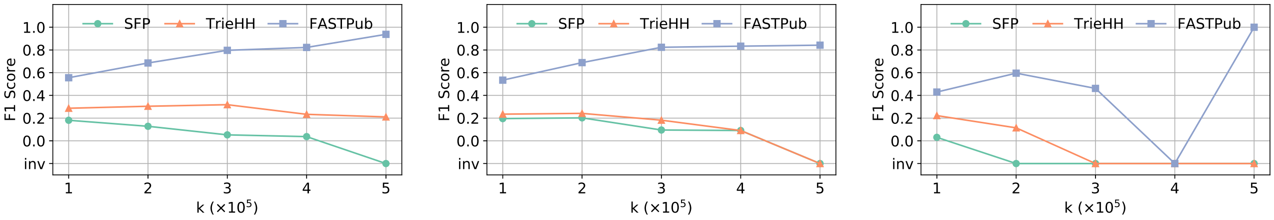
which yields Theorem 4.

Appendix D
Experiments in Oldenburg Dataset

Oldenburg dataset: Compared to the MSNBC dataset evaluated in Section 6. The Oldenburg dataset simulates trajectory publication tasks in large-scale settings with more possible locations and trajectory fragments. The experiment results of frequent sequence mining and count query tasks in the Oldenburg dataset are presented in Figs. 10 and 11, respectively.[10] FASTPub also outperforms the benchmarks in both tasks in the Oldenburg dataset, gaining higher F1 scores and lower relative error. In addition, the algorithms tend to perform badly when the fragment length is 5. Because there are too few fragments satisfying k-anonymity, which increases the difficulty of trajectory publication tasks. SFP performs worse in 4-length queries compared to 3-length queries in count query tasks, but not in frequent pattern mining tasks, due to its trajectory trimming strategy.

In particular, publishing 5-fragments in the Oldenburg dataset with $k = 400,000$ and $k = 500,000$ suffers a strange performance drift. It is due to the low number of fragments satisfying k-anonymity (1 fragment for $k = 500,000$ and 3 fragments for $k = 400,000$). It makes the publication process with high uncertainty and coincidence. The performance when $k = 500,000$ gains a great improvement compared to when $k = 400,000$, because when $k = 500,000$, the number of candidates in the last round becomes lower than C, which forces the noise factor η to be smaller. As a result, the performance is improved compared to $k = 400,000$. Such a phenomenon is likely to happen for any dataset when the number of candidate fragments is small enough due to lacking frequent fragments, high k, or high l_{max}. It can be also circumvented by decreasing C.

[10] Some methods may fail to identify any fragment satisfying k-anonymity in some cases. Since the data utility measures are not available here, they are marked "invalid" (inv) in the figures.

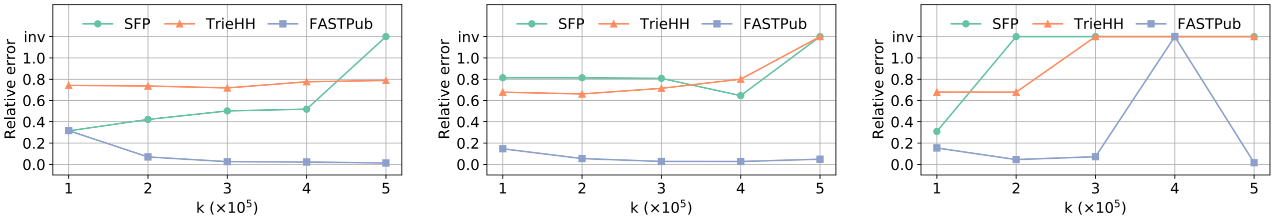


(a) 3-length sequences

(b) 4-length sequences

(c) 5-length sequences

Fig. 10. Performance of frequent sequence mining task in Oldenburg dataset.



(a) 3-length query

(b) 4-length query

(c) 5-length query

Fig. 11. Performance of count query task in Oldenburg dataset.