Understanding the protection of privacy when counting subway travelers through anonymization

# Abstract

Public transportation, especially in large cities, is critical for livability. Counting passengers as they travel between stations is crucial to establishing and maintaining effective transportation systems. Various information and communication technologies, such as GPS, Bluetooth, and Wi-Fi, have been used to automatically measure the movements of people. Regarding public transportation applications, the Automated Fare Collection (AFC) system has been widely adopted as a convenient method for measuring passengers, in particular because it is relatively easy to uniquely identify card owners and as such, the movements of their card holder. However, there are serious concerns regarding privacy infringements when deploying such technologies, to the extent that Europe's General Data Protection Regulation effectively has forbidden straightforward deployment for measuring pedestrian dynamics unless explicit consent has been provided. As a result, privacy-preservation techniques (e.g., anonymization) must be used when deploying such systems. Against this backdrop, we investigate to what extent a recently developed anonymization technique, known as detection k-anonymity, can be adapted to count public transportation travelers while preserving privacy. In the case study, we tested our methods with data from Beijing subway trips. Results show different scenarios when detection k-anonymity can be effectively applied and when it cannot. We also found that it can be challenging to set proper detection k-anonymity parameters. Furthermore, through detection k-annmytiy, it is possible to count travelers between two locations with high accuracy. However, counting travelers from more than two locations lead to more inaccurate results.

Keywords: detection k-anonymity; geoprivacy; preserving privacy; General Data Protetion Regulation; public transportation

# Introduction

Residents and visitors depend on public transportation in cities and towns worldwide. Analyzing public transportation data helps in understanding and improving services. One beneficial use case is counting passengers as they move between locations. Measuring passenger movements has by now become relatively simple for many modern public transportation systems, as users check in and check out of subways and buses using customized smart cards. The information obtained from measuring passengers behavior is essential for overall fleet management and effective transport scheduling, leading to improving the quality and reliability of the public transportation service, identifying travel patterns, or emergency preparedness (Boreiko & Teslyuk, 2016)(Patlins & Kunicina, 2015) (Dunlap et al., 2016), (Wirz et al., 2012)..

In order to measure passenger movements, their locations must be known. The growing use of location-enabled technologies is enabling an increasing number of parties to access this information. Consequently, people are concerned about their geoprivacy. Geoprivacy is *a special kind of information privacy that involves a person's right to decide how, when, and to what extent location data about himself or her is shared* (Drummond et al., n.d.). Unfortunately, many people have only limited knowledge of how the underlying technology for using location information works, such as what can(not) be inferred from an individual's location over time. Asking for a person's consent can therefore be asking something too difficult. We take the standpoint that privacy should be protected upfront such that no consent is needed: no party has, by design, access to a person's sensitive geographical data.

In the case of measuring passenger behavior for public transportation, the main challenge faced is that there is a significant risk of privacy violations when using smart cards . This is because each smart card is individually recognizable. In other words, anonymization of the data contained on a card is not enough. As a consequence, the public needs to trust the organizations that provide the cards and organizations that use their data for further analyses. For this reason, the analyses of the data extracted from the use of such cards are generally strictly regulated by privacy laws, such as Europe's General Data Protection Regulation (GDPR) (Georgiadou et al., 2019)(European Union. 2016).

The current strategies for preserving privacy rely on replacing actual identifiers with pseudonyms, which still allows tracking over time and space. In 2013, the largest Japanese train company announced its intention to sell its passenger dataset to third-party companies (Avoine et al., 2014) (Megan Geuss, 2013); they planned to anonymize the data by replacing sensitive information such as the card-owner's name and residence with an anonymous ID. Obviously, this is not enough for protecting privacy. By simply analyzing patterns of an individual card and combining those patterns with other datasets it has been shown that the identification of an individual person is still possible (Avoine et al., 2014). Numerous studies have been conducted on re-identifying formerly anonymized individuals, and they have shown that it is often not difficult to do so (Fechner & Kray, 2012), (Avoine et al., 2014). It has been demonstrated that auxiliary data can be used to re-identify individuals in datasets that appeared perfectly anonymized on their own (Fechner & Kray, 2012) (Emam et al., n.d.). More is needed. K-anonymity (Wang et al., 2014) and differential privacy (Goos et al., 2008) (Mir et al., 2013) are two of the more common approaches used in the geospatial sciences, with the aim of maximizing the value of dataset containing location information, while minimizing the chances of identifying individuals or groups in the data.

As k-anonymization is one of the most widely used methods for anonymizing all identifiers, we explore this method in this paper. Stanciu et al. (2020) have previously developed a technique based on k-anonymity that effectively ensures that every identifier is converted into a pseudonym that is assigned to at least k-1 other card identifiers. Data cannot be traced back to a single individual using this method; instead, data can be traced back only to a group of k individuals. We call this technique *detection k-anonymity*.  Although privacy-preserving options exist, it is still unclear to what extent detetction k-anonymity is effective in the case of preserving geoprivacy.

Against this backdrop, this paper examines to what extent and under which conditions we can accurately apply detection k-anonymity to counting passengers who travel on a particular subway line while ensuring that data cannot be traced back to an individual. In the case study, we applied detection k-anonymity on the dataset of trips made on the Beijing subway system using smart cards to check in or check out travelers. We used this data as the ground truth to evaluate the balance between privacy preservation and the accuracy when counting trips.

The paper is organized in the following way: the next section gives an overview of the method used to protect privacy. The third section describes the data used in the case study, the fourth section introduces the experimental setup and the findings of the research. The final section concludes the paper.

# Protecting privacy through detection k-anonymity

## Background

There are many methods to measure pedestrian dynamics in public transportation, including manual counting and Automatic Passenger Counting (APC) devices (Patlins & Kunicina, 2015)(Tilg et al., 2021), another method is the Automated Fare Collection (AFC) system. AFC has established a smart card system that is used by metropolitan governments throughout the world to compute prices for various transport lines in cities, such as buses and subways. AFC is the method that the Beijing subway data has been collected.

Another typical data collection tool that has become widely popular is detecting individual mobile devices through the Wi-Fi or Bluetooth signals they transmit. As these signals carry device-identifying information, such as a unique network address, they can, in principle, be used for tracking (Oransirikul et al., 2014). This method is deployed, for example, by Transport for London to monitor how passengers travel through the subway**.**

In this study, our goal is to measure pedestrian dynamics in a subway setting, and the way that privacy preservation by detection k-anonymity works in this setting. For an AFC system every person carries a smart card to use the subway; we have counters that detect the cards when passengers check in or check out, and each card is marked with an identifier that can be read by these counters.

In our approach, we demand that the check-in and check-out counters, which collect identifiers, timestamps, and locations of each card, are responsible for applying anonymization techniques immediately upon detecting a smart card. By collecting card identifiers at each counter during a small timespan, and subsequently replacing such an identifier with a *k-anonymous pseudonym*, the system should, in principle, provide a sufficient degree of privacy.

The description we provide corresponds to the way that we perform privacy preservation in Wi-Fi detection systems, which led to the development of detection k-anonymity.When a device passes a sensor, the device’s identifier, a timestamp, and the sensor’s identifier are logged (we assume that the actual location of the sensor is known). To successfully anonymize devices, we collect data during some time interval referred to as an **epoch** (e.g., 5 minutes). After an epoch has elapsed, we replace each device identifier with a pseudonym such that each pseudonym is used for at least k devices detected during that epoch, and record how many devices have been detected per assigned pseudonym. This information is then sent to a central server.

## Detection K-Anonymity

A privacy-preserving AFC passengers-monitoring environment consists of the following:

* A network of subway lines with each line consisting of a source and a destination, and counters at each source and destination gathering card identifiers; a counter acts as a sensor *s*; all counters form a set *S*.
* A set of *E* of *N* epochs, jointly spanning an elapsed time *T* during which the system runs; we should have enough data during each epoch to be able to apply anonymization.
* A set *IDS* of *M* card identifiers detected by our system during *T*; we assume that each card identifier represents a passenger;.

A detection is a triplet *(id, s, e), id ∈ IDS, s ∈ S, e ∈ E*, representing a card uniquely identified by its identifier *id*, sensed by counter *s* during epoch *e* (Stanciu et al., 2020). Each detected card identifier is first mapped to a 27-bit **pseudonym**, with the *PID* denoting the set of all possible pseudonyms. A pseudonym is derived from a card identifier through secure hashing, establishing that pseudonyms are uniformly distributed in the interval We devise a anonymization procedure *m* to a new set of **multipseudonyms** *MPID*, such that for each detected *pid* *∈ PID* there are at least other detected pseudonyms {,…., } ⊂ *MPID* with *m(pid) = m().* As mentioned, we assume that each counter stores only multipseudonyms; we guarantee that for each stored multipseudonym, a counter detected at least *k* different devices (i.e., pseudonyms) during each epoch. A simple example of such an anonymization procedure is the **truncation operation** *trunc (id, nb)* that removes all but the left most *nb* bits from the binary number *pid*, for all *pid ∈ PID*.

Note that to ensure that at least *k* pseudonyms are mapped to the same multipseudonym, we need to correctly set a value for *nb*. In other words, we should figure out how many bits to keep to ensure detection k-anonymity. If we keep too many bits, truncation of detected pseudonyms may leave us with multipseduonyms for which there are simply less than *k* detected pseudonyms. In that situation, we have no choice but to discard those multipseudonyms. Obviously, this may significantly affect the accuracy of passenger counts.

As an alternative to discarding multipseudonyms (and thus detected pseudonyms), we deploy a systematic method to map *k*-anonymity-disobeying detected multipseudonyms and apply that method to all sensors. We addressed this problem with a **correction method**. Assume there are *n disobeying* multipseudonyms during an epoch. Each such multipseudonym has less than *k* detected pseudonyms. We first sort these multipseudonyms and subsequently keep only the first ones, systematically evenly spreading the counts from the discarded multipseudonyms over the multipseudonyms that we keep. Note that each kept multipseudonym will now have an associated count of at least *k* devices. To illustrate, consider the following five disobeying multipseudonyms sets after truncation by keeping four bits (*nb = 4*) and *k* = 2: {(0011, 1), (0111, 1), (1011,1), (1100,1), (0000,1)}. There is a count of 1 associated with each of these multipseudonyms, which violates the constraint of at least two. To apply the correction, the disobeying multipseudonyms are sorted leading to {0000, 0011, 0111, 1011, 1100}. We then keep only the first entries, namely {0000, 0011}, and evenly spread the counts of the other-disobeying multipseudonyms, leading to the multiset {0000, 0011, 0000, 0011, 0000}, represented as {(0000,3), (0011,2)}.

Once the data at a counter have been anonymized using detection k-anonymity, it is sent to a central server. The central server contains two types of anonymized data obtained from both checked-in and checked-out trips. The data that we have collected so far can now be used as a basis for performing a counting method.

# Data set

In the case study, we adopted the weekday public transit smartcard records in April 2010 of the Beijing subway (Wang et al., 2016) (Zhou et al., 2017) to demonstrate how our methods work. This data set contains 239 728 records that belong to trips that happened during one week. Each record contains a unique card identifier, the day, time, and location at which an individual checked in, and later checked out. The smart cards that passangers use are usually purchased anonymously through resellers or automated machines, and have a unique ID. They are normally unregistered as belonging to a specific individual, so do not carry any personal information about identities. As a result, apparently customers can consider these smart cards completely private, while also keeping some of the benefits of personal travel permits, such as the capacity to be used many times or some offers from transportation companies for smart card holders. These unique smart cards allow us to evaluate the behavior and the number of passengers who travel between stations.

The dataset contains precise information on which card was checked in where, and where it later was checked out. In other words, we have accurate ground truth data on actual passenger dynamics. In this sense, the Beijing dataset is much better for evaluating our anonymization method than possible with Wi-Fi-based datasets. Apart from the fact that Wi-Fi detection is subject to many failures (caused by, for example, interferences, erratic detection and transmission ranges, varying signal strengths, and randomization of MAC addresses), attaining the ground truth is extremely difficult. The latter involves knowing which devices are carried by whom, and subsequently physically tracking an individual.

# Experiments

For our experiments, we simulate two scenarios: (1) counting travelers from one location to another, and (2) counting travelers from two different locations to a common destination. We are conducting this scenario to determine to what extent we can count travelers when they check-in at a location and move straight to a destination (A to B). A second scenario involves adding another source to the common destination (A to Z and B to Z) to determine how counting passengers from two different sources interferes at the common destination, as it may be more difficult to reliably associate a multipseudonym at the destination with the orginal source.

Data for this study is based on a 239 728 trips, that have taken place during a week in April 2010 on the Beijing subway. For our goal, counting the number of devices detected at location A during many successive epochs and later at location B over again a series of epochs, we applied the detection k-anonymity for different values of *k, nb* and different epoch lengths.

First, we consider an isolated line, that is, only those trips that occurred between two specific locations (A to B). For counting the number of trips between these two locations, each counter applies our privacy-preserving algorithm with the same values for all parameters (i.e., *k*, *nb*, and the epoch length); we consider that each counter stores only pseudonyms, and only during the length of an epoch, to subsequently assign pseudonyms to multipseudonyms. After applying detection k-anonymity over epoch *e*, all pseudonyms gathered during *e* are discarded, and the multipseudonyms, along with their respective counts, are sent to a central server.

To associate multiple pseudonyms with a single multipseudonym, we could ideally apply truncation to the original card identifier. However, truncation works well only if we can make the assumption that detected card identifiers are uniformly distributed over the entire possible space of card identifiers. To this end, each counter first applies a globally agreed upon secure hashing function that generates a unique, yet uniform random pseudonym for each detected card identifier. We then apply detection k-anonymity on such pseudonyms to produce multipseudonyms. The uniform distribution of pseudonyms guarantees that when constructing a multipseudonym by truncation, there is no built-in bias toward which multipseudonyms are constructed, nor is there a bias toward the actual number of associated pseudonyms for each multipseudonym. A counter keeps track of how many pseudononyms have been assigned to a single multipseudonym, to later send the pairs *(multipseudonym, number of detections*) to a central server*.*

## Simulated environment for one line

To get a clear understanding of the behavior of the anonymization process, we tested the design on subway trips from Beijing in various settings. As the dataset was rather sparse, we pretended that all registered trips occurred on the same day.

Many parameters shape the experiments, such as values of *k*, the truncation *nb*, and the epoch length. We tested various values for each parameter during our experiment to examine in which situations we still have high accuracy in counting detected devices on a specific line.

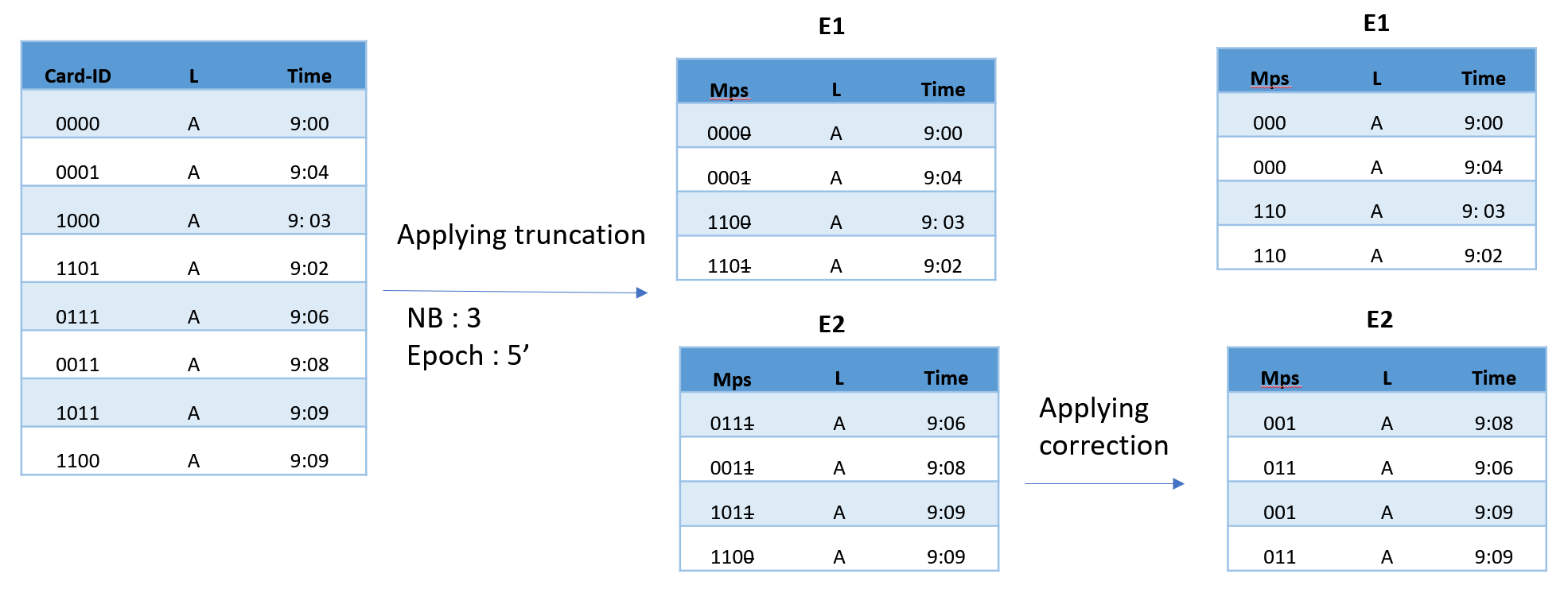


Figure 1. Applying detection 2-anonymity on the example data set with the parameter values of k: 2, nb: 3, and epoch: 5 min (Mpse: multipseudonyms).

In Figure 1, we show how detection k-anonymity works on an example of eight trips between stations A and B. We perform this experiment for every 5 minutes, i.e., with epochs lasting 5 minutes. We keep three bits (*nb = 3*) and set *k* = 2. In our example, we consider two successive epochs of 5 minutes each. Applying truncation during the first epoch (), keeping only the 3 leftmost bits, transforms the detected set of pseudonyms {0000, 0001, 1100, 1101} to the multipseudonyms {(000,2), (110,2)}. In other words, we record that we have detected multipseudonym 000 through two actual pseudonyms. The same holds for multipseudonym 110.

The situation is different for epoch where we have the pseudonyms {0011, 0111, 1011, 1100}. If we apply truncation, we are left with the multipseudonyms {(001, 1), (011, 1), (101, 1), (110, 1)}. Each of these multipseudonyms has an associated count of 1, i.e., each disobeys the constraint that the count should be at least 2. Applying the correction, we keep, after sorting, only the first entries, and evenly spread the counts of the other-disobeying multipseudonyms, leading to the set {(001, 2), (011, 2)}. Note that in this way, we have not lost any counts (the total count during is still 4).

After applying detection k-anonymity at the end of each epoch, the anonymized data is sent to a central server. At the central server, we have two tables containing the data of check-in and check-out locations. The question arises, how do we count the number of people going from one location to another? We do so using a simple matching algorithm: see if a multipseudonym during an epoch at location A has also been recorded during an epoch at location B. The algorithm for matching is shown in Figure 2. After applying detection k-anonymity, but now for illustration purposes with a large value for *nb* (*k=2, nb=8*), we have two tables, one belonging to location A during epoch (*09:00 to 09:05*) and one belonging to location B during epoch (*09:10 to 09:15*). Here, we can incorporate the average travel time into the counting process. This will give us the range of epochs in which we should expect to see the multipseudonyms from A to B. By knowing the average travel time, we can more easily identify relevant departure and arrival epochs, yet strictly speaking, we need not know the length of the trips. In this example, if we consider an average travel time of 10 min, we expect to find multipseudonyms from A to B during epoch (10 minutes after check-in time). We pick the multipseudonym “0000 0000” from table A and start searching to find the same multipseudonym at B; in the first row of table B, we have the same multipseudonym; thus, we match these multipseudonym as a trip that has occurred by the multipseudonym “0000 0000” from A to B, we do the same for other multipseudonyms to find a match for them as well. There may be times when it is impossible to match a multipseudonym; for instance, for the second occurrence of “1100 0001” in table A (which we indicated in bold), there is no match any more for “1100 0001” at table B, which means this multipseudonym will have arrived during another epoch (or possibly, at another station).

Table

Description automatically generated

Figure 2. Matching trips between two location A and B based on the data on the central server, resulting in counting a total of 8 trips.

As mentioned, we first take an isolated line from the Beijing subway data set (M122 to M113) with 545 trips. In Figure 3, the number of checks-in/out is shown during each epoch (A1 means location A, epoch ) in two tables for locations A and B (check-in/out). The travel time for this line is generally between 23 and 30 minutes; we take the epoch length as 10 minutes. Knowing that the average travel time is at least 23 minutes, we expect to see travelers from A start to record at B during epoch. For epochs and at location B, we know that the number of devices is zero because travelers have not yet arrived. The final step is to apply the detection k-anonymity to each epoch.

Table, calendar

Description automatically generated

Figure 3. The number of epochs and the number of devices that were recorded during each epoch for locations A and B (check-in/out).

Figure 4 shows the results of counting detected passengers at location A, and later at location B during different epochs; then, according to the matching algorithm, the multipseudonyms are matched as trips that occurred between these two locations.

We interpret the results as follows: the first row is the number of devices recorded for the ground truth. They checked in during epoch *e* at A and later checked out during epoch *e'* at B (we know the actual number of trips because we have the exact card identifiers including where they checked in and later checked out). In the following, to achieve detection k-anonymity, we kept different numbers of bits to compare the accuracy of our design in various settings with the ground truth. As shown for , when we increase the numbers of bits to keep, we attain higher accuracy in counting the number of trips between two locations. The differences between the results for different compared with the ground truth is shown in Figure 4.

Table

Description automatically generated

Figure 4. The counting result of detection k-anonymity in compared with the ground truth with various settings, k=2, epoch length 10 min, and different number of bits to keep.

To further clarify, consider column A2-B4 (i.e. trips that started at A during epoch and arrived at B during ). We have a known ground truth of 25 trips. For , the algorithm counted 70 trips, which is considerably higher than the ground truth. We know that 45 out of these 70 trips actually arrived at B during other epochs than (). The reason for our large number of counts is that for , because of truncation, we have only four multipseudonyms (00, 01,10,11) in A and B. Using detection k-anonymity, we have the multiset {(00:16), (01:19), (10:28), (11:25)} at A and {(00:21), (01:27), (10:22), (11:13)} at B. Our algorithm matched all these multipseudonyms in source and destination; for example for multipseudonyms "00," we have 16 of them at A and 21 at B, so the algorithm counts 16 (smallest value) trips made by multipseudonym "00" from A to B, and it repeats the same procedure with the other three multipseudonyms, leading to a total of trips.

In the case of , the algorithm counted 22 trips from A2 to B4, which is closer to the ground truth than for . All bits, and thus pseudonyms, were retained at A and B, so only the correction step of detection k-anonymity was applied. Once k-anonymity was established, the matching algorithm began looking for multipseudonyms from A at B. The matching algorithm found only 22 trips out of 25; half of the original multipseudonyms have been replaced with other, smaller multipseudonyms. Smaller, because we first sort all multipseudonyms and effectively keep only the smallest ones for matching.

In order to demonstrate accuracy, we counted true positives, false positives, false negatives, and true negatives. We define false and true counts as follows:

*True positives*: the number of trips we were able to count that actually occurred.

*False positives*: the number of trips we counted that actually did not occure.

*True negatives*: the number of trips we did not count, and that indeed did not happen.

*False negatives*: the number of trips we did not count, but actually occurred (i.e., we missed them).

Table

Description automatically generated

Table

Description automatically generated

Figure 5. counting the number of trips for k=2 and four different nb, besides the number of true positives, false positives, false negatives, and true negatives.

The four tables in Figure 5 show the accuracy of our counting method for four different values of and . For each table, the first column shows epochs for which we want to count how many people moved from A and arrived at B during those epochs; the second column shows the actual number of trips that happened (i.e., the ground truth), to which the results of our matching algorithm in the third column are compared with. Other columns include true positives, false positives, false negatives, and true negatives.

The sum of *TP* and *FP* column values is equal to the number of trips that our method counted (N-Trips column), where *TP* indicates the number of trips that our algorithms counted correctly. False positive represents the number of trips that were mistakenly recorded but did not occur, and these trips do not appear in *GT*.

As we mentioned before, when only truncation is performed, such as with , the number of different multipseudonyms is low at both locations. A small number of multipseudonyms leads to incorrectly matching many trips that actually did not happen, leading to many false positives. Figure 5 for the table with illustrates this. The number of false positive trips decreases when the number of bits is increased, shown in the other tables. By increasing the number of bits to keep we have more different multipseudonyms, leading also to more corrections (because multipseudonyms do not have enough associated trips). When applying corrections, we lose trips as false negatives.

In the correction phase, as we explained before, for having the detection k-anonymity some multipseudonyms will be replaced by others (but allowing counts greater or equal to *k*). As a consequence, we lose some multipseudonyms, and those multipseudonyms are then no longer available to match with arrivals or departures. In fact, lost multipseudonyms are matched by using their replacements, and thus lead to counting trips that did not take place. These inaccurate matches count as false positives. At the same time, each lost multipseudonym will also mean that we miss trips, leading to false negatives.

Due to the connection between *FPs* and *FNs*, the best scenario is that, all the *FPs* will be compensated by an equal number of *FNs*. In other words, we can achieve the highest level of accuracy by having only a correction phase (and having sufficient data). The most accurate counting occurs when false positives and false negatives are equal or close to each other. In the table for , for the last row (*A6-B8*)*,* the *GT* is 24; the algorithm found 24 trips during this epoch, from these 24 trips 12 of them counted correctly as *TP* and 12 of them should be *FP*, so if 12 of them are false positive, then ideally, the same value should be the number of trips that really happened but our algorithm did not count (*FN*). We have in this row, so that is the best we can attain.

The value for *TN* is the same in all tables regardless of the detection k-anonymity setting; the reason is that the algorithm cannot count a trip that does not occur between two locations. Therefore, it does not matter how the algorithm is set; the trips that have not happened can not be counted.

For an accurate count of how many people left during one specific epoch and their arrival at the destination, we must consider several consecutive epochs at the destination. This is because we do not know precisely during which epoch a passenger will actually check out. For example, In Figure 3, 107 passengers checked in during epoch while only 38 passengers appeared during epoch at B. We can conclude that passengers who left at A during may have arrived at B during other epochs than To test this assumption, for each departure epoch at A, and expected arrival epoch at B, we also looked at the next epoch at B and checked potential arrivals from A who left during . In Figure 6, we displayed this as follows. For epoch at A (A1), we looked at arriving multipseudonyms at B during epochs as well as (denoted as B3+B4). In this way, we found a higher number of multipseudonyms from A at B. We repeated this approach for other departure/arrival epochs.

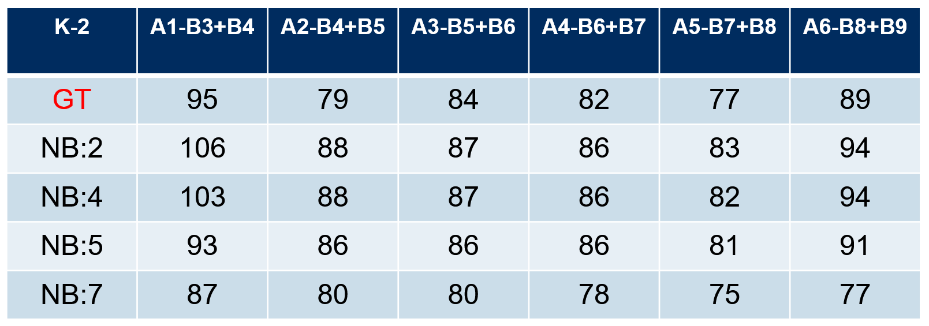


Figure 6. Counting the number of multipseudonym that appear at each epoch of A and later in two epochs of B.

We also looked at the effect of the epoch length, as shown in Figure 7. Notably when epochs are small compared to the expected travel time, the number of travelers that the matching algorithm counts is closer to the ground truth than with large epochs. This happens when is very low, so all the theoretically existing multipseudonyms are used at both source and destination. In addition, larger epochs have a higher chance of having travelers from earlier epochs, as we witnessed before (Figure 6), so we should indeed see more false positives.

Table

Description automatically generated

Figure 7. Comparison results for different epoch lengths (10 min, 15 min) from A to B, nb=2, k=2.

Large epochs mean we have more data for applying detection k-anonymity in comparison to having small epochs, so truncation allows us to keep more bits and requires less correction. However, having large epochs (relative to the time a trip takes) also means that correctly matching departures to arrivals becomes more difficult. When applying detection k-anonymity, there is a higher risk of losing identifiers because of fewer data per epoch. If sufficient detections can be guaranteed, epoch lengths become less critical.

## Simulated environment for combining trips

Matters may quickly get out of hand when combining trips. First, we consider the situation of counting passengers moving from A to Z, with a counter at location A gathering a set of pseudonyms PIDA, mapping them to the multiset MPIDA. At destination location Z, we have PIDZ and MPIDZ, respectively. Now consider the situation that we have another source, B, representing people who travel from B to Z, resulting in a set of pseudonyms PIDB and multipseudonyms MPIDB, respectively, for counter B. The counter at Z detects through PIDZ precisely the passengers moving from A to Z and B to Z, respectively. However, there are situations when the multipseudonyms in MPIDZ, corresponding to travelers from A, will have been "contaminated" by travelers who moved from B and arrived at Z. Let us look at this situation.

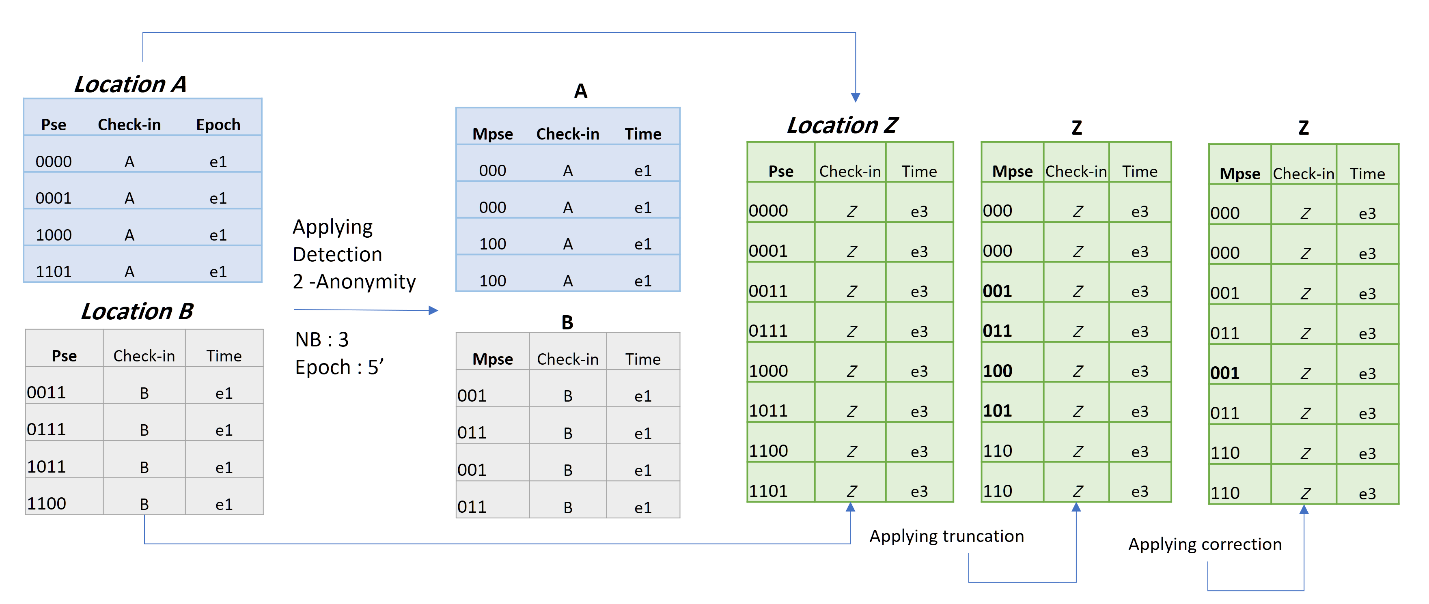


Figure 8. Applying detection K-Anonymity to three different locations with a common destination Z. Pse: pseudonyms, Mpse: multipseudonyms.

Figure 8 shows three stations: A, B, and Z; people moved from A and B to later arrive at Z. Based on this example, we want to show how pseudonyms from A at Z are contaminated by travelers who moved from B to Z. At all locations, and for each epoch, we applied detection k-anonymity on locations with and . For location A and epoch , truncation alone was enough to reach 2-anonymity; for B (and again, ), additional correction was needed. After passengers arrive at Z, detection k-anonymity resulted in what is shown in green. After truncation, we have four disobeying multipseudonyms: {001, 011, 100, 101}. We first sort these and apply the correction method. After sorting, the multipseudonyms {001, 011} were used to correct 100 and 101, now leading to {(001,2), (011,2)}. However, note that multipseudonym 100 was originally from location A, while multipseudonym 001 originated from B. In other words, the correction yields that we will wrongfully match a trip from A to one coming from B. This mismatch is entirely due to mixing trips from B with those from A: the trips from B are said to contaminate those from A.

To further analyze this situation, we gradually add travelers coming from B (called joiners) to the travelers arriving at Z and coming from A. When keeping only few bits when applying truncation, we have the lowest accuracy compared to the ground truth because travelers from A may count as travelers from B to Z. This situation is sketched in Figure 9. For detection 2-anonymity in this line, we kept all bits (27 bits) and we see results are generally better for counting travelers from A to Z when we have all joiners (all travelers from B) (100 %) Since we kept all bits, the algorithm matched multipseudonym accurately.

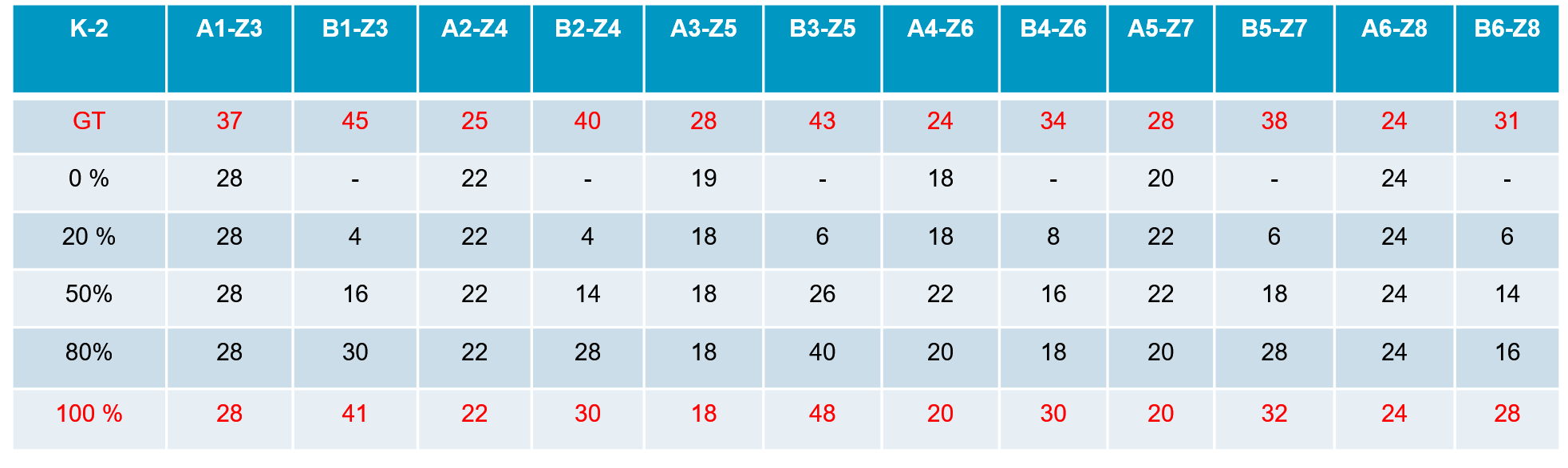


Figure 9. constantly added travelers who moved from B to Z to travelers from A to Z.

In Figure 10, we display the results for different values of *nb*, *k=2,* and combining each epoch of Z with thenext epoch*,* for counting the number of trips between two lines A to Z and B to Z. We can see when we choose an acceptable epoch length and many bits to keep, we can obtain high accuracy in counting the number of passengers who move from one location to another even if we mix with another departure station, like B. However, setting the correct parameters for epoch length and truncation sizes have proven to be complicated.

It can be seen in Figure 10, that by increasing the number of bits, the results come close to the ground truth, implying a higher degree of accuracy. By keeping all bits and having only the correction phase the algorithm was able to count travelers with high accuracy for both lines A to Z and B to Z, which we show in Figure 11.

A screenshot of a computer

Description automatically generated with low confidence

Figure 10. The number of trips during each epoch from A to Z and B to Z compared to GT.

Table

Description automatically generated

Figure 11. Comparison nb=2 with nb=27 for two lines with K=2.

# Conclusion

Sustainable city planning relies heavily on the counting of passengers in public transportation systems. Tracking passenger flows can be done in many ways, yet all these approaches have their drawbacks. A prevailing concern is the preservation of privacy. The present study was designed to determine the effect of preserving privacy when counting subway travelers. Our objective was to assess the extent and conditions under which detection k-anonymity can accurately count subway passengers while ensuring that individuals cannot be traced from the final data set. In the current study, comparing the results of our algorithm with the ground truth showed that passengers between two locations can be counted accurately if the detection k-anonymity algorithm is properly configured. However, when combining trips from several departure stations yet having the same destination, results quickly get worse. This is mainly caused by the inability to correctly match multipseudonyms, as a single multipseudonym at the destination may actually have been constructed from trips coming from both origins.

This finding answered the questions of other studies in this area that it is possible for public transportation companies to record and count passengers and protect the privacy of individuals. We showed that it is possible to use anonymization techniques that prevent tracing back to an individual. It should be noted, however, that in some situations applying privacy preservation may lead to a serious decrease in the accuracy of counting. In future work, we plan to explore more recent alternative techniques that may lead to protecting privacy at higher accuracies.

# References

Asadpour, M., & Dashti, M. T. (2011). A privacy-friendly RFID protocol using reusable anonymous tickets. *Proc. 10th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications, TrustCom 2011, 8th IEEE Int. Conf. on Embedded Software and Systems, ICESS 2011, 6th Int. Conf. on FCST 2011*, 206–213. https://doi.org/10.1109/TrustCom.2011.29

Avoine, G., Calderoni, L., Delvaux, J., Maio, D., & Palmieri, P. (2014). Passengers information in public transport and privacy: Can anonymous tickets prevent tracking? *International Journal of Information Management*, *34*(5), 682–688. https://doi.org/10.1016/j.ijinfomgt.2014.05.004

Bettini, C., Jajodia, S., Samarati, P., & Wang, X. S. (n.d.). *PiLBA’08 Privacy in Location-Based Applications Workshop co-located with ESORICS 2008*.

Boreiko, O., & Teslyuk, V. (2016). Structural model of passenger counting and public transport tracking system of smart city. *Perspective Technologies and Methods in MEMS Design, MEMSTECH 2016 - Proceedings of 12th International Conference*, 124–126. https://doi.org/10.1109/MEMSTECH.2016.7507533

Drummond, J., Billen, R., Joao, E., & Forrest, D. (n.d.). *Dynamic and Mobile GIS Investigating Changes in Space and Time*.

Dunlap, M., Li, Z., Henrickson, K., & Wang, Y. (2016). Estimation of origin and destination information from bluetooth and wi-fi sensing for transit. *Transportation Research Record*, *2595*, 11–17. https://doi.org/10.3141/2595-02

Emam, K. el, Jonker, E., Arbuckle, L., & Malin, B. (n.d.). *A Systematic Review of Re-Identification Attacks on Health Data*. https://doi.org/10.1371/journal.pone.0028071

European Union. 2016. (2016). *General Data Protection Regulation (GDPR)*.

Fechner, T., & Kray, C. (2012). *Attacking Location Privacy: Exploring Human Strategies*.

Georgiadou, Y., de By, R. A., & Kounadi, O. (2019). Location privacy in the wake of the GDPR. *ISPRS International Journal of Geo-Information*, *8*(3). https://doi.org/10.3390/ijgi8030157

Goos, G., Hartmanis, J., Van, J., Board, L. E., Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Kobsa, A., Mattern, F., Zurich, E., Mitchell, J. C., Naor, M., Nierstrasz, O., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., & Weikum, G. (2008). *Theory and Applications of Models of Computation*.

Maltoni, N., Magnani, A., & Calderoni, L. (2020, August 25). Privacy threats in low-cost people counting devices. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3407023.3409195

Megan Geuss. (2013). *Japanese railway company plans to sell data from e-ticket records*. https://arstechnica.com/information-technology/2013/07/japanese-railway-company-plans-to-sell-data-from-e-ticket-records/

Mir, D. J., Isaacman, S., Caceres, R., Martonosi, M., & Wright, R. N. (2013). DP-WHERE: Differentially private modeling of human mobility. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*, 580–588. https://doi.org/10.1109/BigData.2013.6691626

Oransirikul, T., Nishide, R., Piumarta, I., & Takada, H. (2014). Measuring Bus Passenger Load by Monitoring Wi-Fi Transmissions from Mobile Devices. *Procedia Technology*, *18*, 120–125. https://doi.org/10.1016/J.PROTCY.2014.11.023

Patlins, A., & Kunicina, N. (2015). The new approach for passenger counting in public transport system. *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015*, *1*, 53–57. https://doi.org/10.1109/IDAACS.2015.7340700

Stanciu, V. D., van Steen, M., Dobre, C., & Peter, A. (2020). K-anonymous crowd flow analytics. *ACM International Conference Proceeding Series*, 376–385. https://doi.org/10.1145/3448891.3448903

Tilg, G., Pawlowski, A., & Bogenberger, K. (2021). The impact of data characteristics on the estimation of the three-dimensional passenger macroscopic fundamental diagram. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2021-September*, 2111–2117. https://doi.org/10.1109/ITSC48978.2021.9564430

Wang, M., Zhou, J., Long, Y., & Chen, F. (2016). Outside the ivory tower: Visualizing university students’ top transit-trip destinations and popular corridors. In *Regional Studies, Regional Science* (Vol. 3, Issue 1, pp. 202–206). Routledge. https://doi.org/10.1080/21681376.2016.1154798

Wang, Y., Long Xie, ·, Zheng, B., Lee, K. C. K., Wang, Y., Xie, · L, Zheng, · B, & Lee, K. C. K. (2014). High utility K-anonymization for social network publishing. *Knowl Inf Syst*, *41*, 697–725. https://doi.org/10.1007/s10115-013-0674-2

Wirz, M., Franke, T., Roggen, D., Mitleton-Kelly, E., Lukowicz, P., & Tröster, G. (2012). Inferring crowd conditions from pedestrians’ location traces for real-time crowd monitoring during city-scale mass gatherings. *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, 367–372. https://doi.org/10.1109/WETICE.2012.26

Zhou, J., Wang, M., & Long, Y. (2017). Big data for intrametropolitan human movement studies: A case study of bus commuters based on smart card data. *International Review for Spatial Planning and Sustainable Development*, *5*(3), 100–115. https://doi.org/10.14246/irspsd.5.3\_100