Privacy-Preserving t-Incidence for WiFi-based Mobility Analytics

Mohammad Alaggan, Mathieu Cunche, Marine Minier

Privacy-Preserving t-Incidence for WiFi-based Mobility Analytics

Mohammad Alaggan[1], Mathieu Cunche[1], Marine Minier[1]

Univ Lyon, Inria, INSA Lyon, CITI, F-69621 Villeurbanne, France
mohammad.alaggan@inria.fr, mathieu.cunche@inria.fr,
marine.minier@insa-lyon.fr

Abstract. Physical mobility analytics have gained attention lately. As people become more equipped with ubiquitous wireless-communication-enabled mobile appliances, they tend to leave signatures of their presence wherever they go. One particular example is Wi-Fi enabled devices which continuously send packets (called "probe requests") to access points around it even if no connection is established between them. Aggregating a list of such probe requests over a number of geographically distributed monitoring nodes gives rise to a rich set of physical mobility analytics such as visitor density in rush hours and most frequently taken routes.
However, privacy of individual users is a grave concern. To address this concern we propose to implement physical mobility analytics using a collection of privacy-preserving primitives of set operations. The sets are the MAC addresses of the devices observed by one monitoring node. There is at least one set per monitoring node. An monitoring node may have more than one set if the MAC addresses are split according to the time of reception. The primitives we propose are the t-incidences of these sets. We present an ϵ-differentially pan-private algorithm to estimate the t-incidence of n sets, up to multiplicative error $O(\alpha)$, given three $(\epsilon/3)$-differentially pan-private Bloom filters for each of those sets.

1 Introduction

The ubiquitous use of portable devices equipped with radio communications interfaces has enable the passive tracking of their owner. This is especially the case with Wi-Fi enabled devices, which are nowadays commonplace and that are periodically broadcasting a unique identifier (the MAC address) even if they are not connected to a network. This has led to the development of various commercial applications such as physical analytics which are leveraging those radio signals to measure the human activity in a given area. Physical analytics can be beneficial for many applications but also raise privacy concerns [17,12]. Industry have proposed methods [16] to mitigate those issues, but they fail short at protecting individuals' privacy [8]. This lack of strong privacy guarantee are conflicting with privacy legislation and trigger mistrust of the population.

Physical analytics systems are generally composed of several Wi-Fi receivers distributed over a geographical area of interest that receive probe requests from mobile devices (smart phones, tablets, and laptop) of people within this area. Estimating the number of unique MAC addresses seen by an monitoring node during a particular period of time is one kind of mobility analytic, enabling the estimation of visitor density

and rush hours, for instance. More interesting is combining the observations of multiple monitoring nodes. For example, if we know that a particular MAC address was seen by monitoring node 1 at time 10:00 and then by monitoring node 2 at time 10:30, and later by monitoring node 3 at time 11:00, we can conclude that the mobility trajectory of the owner of this MAC address was 1-2-3. We are not interested in statistics about individuals, but rather in collective behavior of many individuals, like the most taken route between points A and B, or whether a particular route is traversed more in one direction than the other.

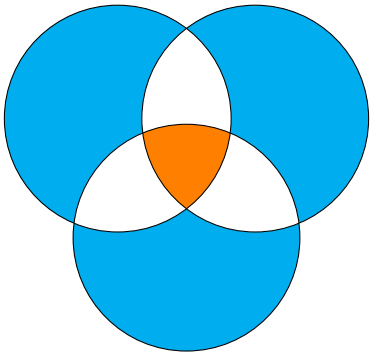

Fig. 1. Example of t-incidences. The three circles represent three sets. The blue (resp. white, orange) area represent the items of the 1-incidence (resp. 2-incidence, 3-incidence) set.

This kind of analytics may be obtained from a set of simpler operations on the sets of MAC addresses observed by n different monitoring nodes, namely the t-incidence. The t-incidence of n sets is the *number* of elements that appear in *exactly* t sets out of n. An example is shown in Figure 1 and in Table 1. Each element may appear in different sets, but they would be exactly t. For instance, the 1-incidence of a set is the number of elements in this set (*i.e.*, its cardinality), the 1-incidence of two sets is the number of elements in their symmetric difference, and the 3-incidence of three sets is the cardinality of the set intersection of all three sets. The sum of t-incidences from $t = 1$ up to $t = n$ is the union cardinality. To implement these set operations we will use a privacy-preserving variant [2,3,1] of the Bloom filter representation of sets [5] because of both its time/space efficiency and privacy guarantees.

Items	a	b	c	d	e	f	g	h
Set A		1			1	1		1
Set B			1		1		1	1
Set C				1		1	1	1
1-incidence set		1	1	1				
2-incidence set					1	1	1	
3-incidence set								1

Table 1. Example of t-incidences. The universe of all items is shown in the first row. The next 3 rows represent sets containing items from that universe of items. A cell contains 1 if and only if the set of its row contains the item of its column. In the last 3 rows we see the t-incidence sets. The t-incidence is simply the cardinality of the respective set, hence the 1-incidence (resp. 2-incidence and 3-incidence) is 3 (resp. 3 and 1). It is worth noting that the t-incidence sets are always disjoint, and their union is the union $A \cup B \cup C$. Consequently, the sum of the cardinalities of the t-incidence sets is the union cardinality $|A \cup B \cup C|$. We can also observe that the 3-incidence set is the intersection $A \cap B \cap C$. The 0-incidence set is only defined with respect to the universe of all items, representing all items which did *not* occur at all. This case is uninteresting for our purposes so we do not consider it here.

1.1 Why t-incidences?

We can use t-incidences to compute cardinalities of any combination of n sets of interest, as long as this combination is symmetric on those sets. Such cardinalities may thus be used for mobility analytics. For example, the sum of all t-incidences for $1 \leqslant t \leqslant n$ is the union cardinality, which can be used to estimate the number of people in a geographical region covered by many Wi-Fi monitoring nodes: the union cardinality of all MAC addresses seen by those monitoring nodes. Further more n-incidence of n sets is their intersection cardinality, which can be used to estimate the trajectory of the user along a path of Wi-Fi monitoring nodes [14], given that the successive sets form a temporal sequence. Alternatively, the n sets may belong to the same monitoring node but at a temporal sequence. In this case we may be able to know how long a user stayed in the same location. For values of t other than 1 and n also prove useful in case we want some sort of robust analytics. For example, if the path taken by the user spans 7 Wi-Fi monitoring nodes, we may take the 6-incidence into account in addition to the 7-incidence to account for the fact that some users may for some reason have only been observed by 6 out of the 7 Wi-Fi monitoring nodes. Some of the reasons may include but are not limited to 1) one Wi-Fi monitoring node along the path was rebooting as the user was passing or 2) the user passed too quickly by this one monitoring node that the phone didn't have a chance to submit a probe request or 3) the user took a small diversion that swayed him away from one of the monitoring node but on the long run she was still on the same end-to-end trajectory. Equivalent examples exist if we consider a temporal trajectory instead. For larger n, like $n = 10$, we may also wish to consider giving more slack by considering all t-incidences greater than or equal to 8 for the trajectory. Consider instead $t = 2$ or $t = 3$. One example why we might be interested in them is the example of a shopping mall. The marketing campaign of the mall may like to know how

many shops are visited by the average user, and specifically target the audience visiting a small number of shops (2 or 3). To know the efficacy of their campaign they would like to observe a decrease in 2-incidence and 3-incidence with and equivalent increase in 4-incidence and 5-incidence.

2 Background

2.1 Bloom Filters

A Bloom filter [5] is a probabilistic data structure for set membership queries. It is composed of a bit vector of m bits, initially initialized to 0, and k random hash functions $h_1, h_2, \ldots, h_k$. To add an item i to the Bloom filter the bits $h_1(i), h_2(i), \ldots, h_k(i)$ are set to 1. And item i exists in the Bloom filter if the bits $h_1(i), h_2(i), \ldots, h_k(i)$ are all set to 1. Items cannot be removed and there is a small probability of false positives: reporting that an item exists while it actually does not, due to hash collisions. The density of a Bloom filter is equal to its Hamming weight (the number of bits set to 1) divided by m.

2.2 Differentially Pan-Private Streaming Algorithm

Definition 1 (Differential Privacy). *[9,10] A randomized function $f : \{0,1\}^m \to \{0,1\}^m$ is ϵ-differentially private if for all $x, x', y \in \{0,1\}^m$*

$$\Pr[f(x) = y] \leqslant \exp(\epsilon \|x - x'\|_H) \Pr[f(x') = y] \ ,$$

in which $\| \cdot \|_H$ is the Hamming distance, and the probability is taken over all the randomness of f.

Definition 2 (Differential Pan-Privacy). *[11] Let* $\mathbf{Alg}$ *be an algorithm that incrementally receive its input (the input is called a* stream *in this case). Let I be the set of all possible internal states of* $\mathbf{Alg}$*, and σ the set of all possible outputs, in which the output is only produced after all input has been consumed. Then for all integers $d \geqslant 1$, the algorithm* $\mathbf{Alg}$ *mapping input prefixes to the range $I^d \times \sigma$ is* ϵ-differentially pan-private against d intrusions *if for all sets $I'_1, \ldots, I'_d \subseteq I$ and $\sigma' \subseteq \sigma$ and for all pairs of adjacent input stream prefixes S and S' (adjacent means the streams are identical except for all occurrences of at most one item):*

$$\begin{aligned} &\Pr[\mathbf{Alg}(S) \in (I'_1, \ldots, I'_d, \sigma')] \\ &\leqslant \exp(\epsilon) \Pr[\mathbf{Alg}(S') \in (I'_1, \ldots, I'_d, \sigma')] \end{aligned}$$

in which the probability is taken over the randomness of $\mathbf{Alg}$.

A ϵ-differentially private variant of Bloom filters (*cf.* Definition 1) was introduced by Alaggan, Gambs, and Kermarrec [2]. The idea was that after all items have been added to the Bloom filter each bit will be probabilistically flipped with probability $1/(1 + \exp(\epsilon/k))$. Algorithms for estimating the intersection cardinality of a differentially private Bloom filter and a set, and of two differentially private Bloom filters have

been proposed in the literature [2,3]. More recently, Alaggan, Cunche and Minier [1] proposed an version with the stronger guarantee of ϵ-differential pan privacy. The main advantage of differential pan privacy is that if the Bloom filter was build incrementally from an input stream of items, then during the extended period of time since the initialization of the Bloom filter until its release the internal state of the Bloom filter kept in the local memory is differentially private against an intruder committing up to d announced intrusions. The same paper also provided an algorithm to compute the t-incidence of n differentially pan-private Bloom filters, and is considered the basis of the current paper.

2.3 Physical analytics based on Wi-Fi

Physical analytics systems relies on the radio signals emitted by portable communicating device to detect the presence and track the movement of individuals. The most commonly used radio technology is Wi-Fi because it is embedded and activated in large number of portable devices and because it genuinely leaks identifying information [13]. Indeed, Wi-Fi device periodically broadcast short messages called probe requests in order to discover surrounding access points. Those probe requests contains the MAC address of the device in clear-text and are typically sent several times per minutes. The MAC address is a globally unique and stable identifier which is allocated to every Wi-Fi device. Because of the open nature of the Wi-Fi communications it is easy to monitor the radio channels in order to collect those identifiers.

A set of monitoring nodes distributed over a geographical area of interest can collect the probe requests from mobile devices (smart phones, tablets, and laptop) of people within this area [15]. Those monitoring nodes are typically small computers equipped with a Wi-Fi interface working in monitoring mode that can capture all the traffic on a specific channel. Such monitoring nodes can collect the unique identifier of each Wi-Fi device in range. The collected data is typically stored in a central server under the form of tuples including the device MAC address (the device identifier), a time stamp, a location identifier (e.g. the identifier of the monitoring node). This data can then be processed to analyze the human activity in the area for instance by counting the number of visitors, the duration and length of visit, etc. Such systems are used in brick and mortar point of sale to better understand customers habits [7] and are also used to study mobility patterns to assist urban planning [15].

3 Our Contribution

Given many differentially pan-private Bloom filters of different sizes, all of which represent the same set, Dwork, Naor, Pitassi, Rothblum, and Yekhanin [11] provided a way to estimate the 1-incidence of this set (its cardinality) up to a multiplicative error $O(\alpha)$. They built upon a primitive they introduced that can can give an estimate of the true density d of a differentially pan-private Bloom filter up to additive error α^2. They showed that if one of the Bloom filters whose size is m had true density d close to α, then md is within $O(\alpha)$ multiplicative error from the desired 1-incidence[1].

[1] Provided that the hash function used was at least 4-independent.

Their idea is about the size m of the Bloom filter. The greater m is, the higher the additive error. The less m is, the higher the number of hash function collisions and consequently the greater the under counting and higher the multiplicative error. The density being as small as α along with 4-independence is a way to guarantee that the collisions and consequently the under counting is kept small. The density being as big as α is a way to guarantee that the size of the bit vector is small, keeping the additive error small.

3.1 Contribution 1: Selecting the Bloom Filter Size m

So how to choose m such that the density will be around α? In [11] they propose to try all sizes that are power-of-two. In particular if the universe contains N MAC addresses, then $\log_2 N$ bit vectors of sizes $m \in \{2^0, 2^1, \ldots, 2^{\lceil \log_2 N \rceil}\}$ will be computed. This approach has several disadvantages. One of them is that the privacy budget will either be too big. If the differential pan-privacy parameter for each Bloom filter is ϵ, then the total privacy budget around $\epsilon \log_2 N$, which is bad for privacy. If instead as [11] suggest we give each of the Bloom filters a small privacy budget, like $\epsilon / \log_2 N$, the error for each will be too high that the estimation method may not converge at all; as confirmed by our experiments. Another disadvantage is that the biggest bit vector will have size N, which may be prohibitive[2]. We address these issue and we additionally generalize their method to more than one Bloom filter, namely to estimate t-incidences on n sets. One way to do this is to use the fact that we have a concrete application and that we can provide a way to choose the size m instead of trying all possible sizes.

If the Bloom filter uses one fully independent hash function, then its density is approximately $1 - (1 - 1/m)^s$, in which s is the size of the set that is encoded in it. Therefore if we have rough expectation of s then we can guarantee that its true density is close to α by setting

$$m = 1/(1 - (1 - \alpha)^{1/s}) \approx s/\alpha \tag{1}$$

as shown in Algorithm 1. The case when we consider $n > 1$ Bloom filters and thus incidences $t > 1$ is different (*cf.* Section 5.2.

3.2 Contribution 2: Extension to More Than One Set ($n > 1$)

We extend the notion of Bloom filter density to deal with this case into what we call t-*density* (*cf.* Definition 3). We can see from Figure 4 and Figure 5 (*cf.* Section 4) that we can also set m to guarantee that the t-density is close to α if we have a rough estimate of the expected t-incidence (*cf.* Section 5.2 for details).

Definition 3 **(t-density).** *The t-density of n sets is the density of the Bloom filter encoding the t-incidence set (*i.e.*, the set of elements appearing in exactly t sets out of those n sets). The definition is relative to the parameters of the Bloom filter in question, i.e. the hash functions and the size m.*

[2] Sampling is used in [11] to address this issue but we do not use this approach since our datasets may contain a very small fraction of the universe of all MAC addresses, in which sampling may throw away most points.

We will also make use of very recent work [1]. Given differentially pan-private Bloom filters of the same size of n sets, Alaggan, Cunche, and Minier [1] provided a way to estimate the t-incidence of these n sets, up to an *additive* error. Since the t-incidences may be very small for large t, we extend their work to guarantee a *multiplicative* error instead. In particular, we will be using their method as a black box in similar spirit to how [11] proceeded; by observing the relationship between t-density and multiplicative error on the t-incidence. To do this we run density estimation algorithm from [1] on three different sizes of the Bloom filters, $2^{\log_2(m)-1}, m, 2^{\log_2(m)+1}$, in which m is the value recommended Algorithm 1 (Note that this algorithm is presented for $t = 1$. For values of $t > 1$ only the function EXPECTEDBLOOMDENSITY will need to be modified as described in Section 5.2). The additive error is set to be α^2. For each t, we then choose the set of Bloom filters whose estimated t-density was closest to α and report the multiplicative error on the t-incidence. We experimentally evaluate this approach in Section 4.

Algorithm 1 Compute Bloom Filter Size

```
1:  procedure ExpectedBloomDensity(s, m)
2:      return 1 − (1 − 1/m)^s
3:  end procedure
4:  procedure ComputeBloomSize(s, α)
5:      Set d ← ∞
6:      for m in {2^0, 2^1, ...} do
7:          Set d' ← ExpectedBloomDensity(s, m) − α
8:          if d' < 0 then
9:              if |d'| > |d| then
10:                 return previous m
11:             else
12:                 return this m
13:             end if
14:         end if
15:         Set d ← d'
16:     end for
17: end procedure
```

4 Experimental Evaluation

We use the Sapienza dataset [4] to evaluate our method. The Sapienza dataset is a real-life dataset composed of wireless probe requests sent by mobile devices in various locations and settings in Rome, Italy. We only use the MAC address part of the dataset, as typical physical analytics systems do. It covers a university campus and as city-wide national and international events. The data was collected for three months between February and May 2013, and contains around 11 million probes sent by 160,000 different devices (different MAC addresses). The released data is anonymized. The dataset

contains 8 setting called POLITICS1, POLITICS2, VATICAN1, VATICAN2, UNIVERSITY, TRAINSTATION, THEMALL, and OTHERS. Each setting is composed for several files. Files are labeled according to the day of capture and files within the same setting occurring in the same day are numbered sequentially. In our experiments we set the parameter $n \in \{1, 2, \ldots, 5\}$, indicating the number of sets we want to experiment on, and we choose a setting, say VATICAN1. Then we pick n random files from the VATICAN1 setting and proceed to estimate their t-incidence according to our algorithm, only if the t-incidence for this subset is nonzero for all t, since the multiplicative error is undefined if the true t-incidence is zero.

In Figure 3 we see that for $n \in \{1, 2, \ldots, 5\}$ sets, we obtain very good multiplicative error on the 1-incidence. The results are average across all settings. Figure 2 average across all n and shows instead the multiplicative error against the true 1-incidence value that is to be estimated. Both figures show that the best performance appears to happen for $\alpha = 0.02$, while we should expect it to happen for $\alpha = 0.002$ instead. This counter intuitive behavior happens because of the privacy parameter $\epsilon = 3$. It is indeed the case that the performance is better for $\alpha = 0.002$ for $\epsilon = 9$ (this experiment not presented here). The problem is that when ϵ is small, effectively ensuring higher privacy and thus requires higher amount of noise, solving for small values of α becomes harder, in the sense that the linear program from [1] becomes unlikely to satisfy. This is a calibration problem that should be addressed: how to choose α given ϵ. In the rest of this subsection we try to explore the performance of the more challenging case of t-incidence for $t > 1$. We will see that the performance depends on some characteristics of different settings.

In Figure 4 and Figure 5 we explore the relation between the size m of the Bloom filter whose t-density is closest to α, and the t-incidence. Each point in Figure 4 has two coordinates, x and y. The y coordinate indicates the Bloom filter size m whose t density is closest to α, out of $\{2^0, 2^1, \ldots\}$. The x coordinate shows the corresponding t-incidence. The black lines highlight the equation x/α for $\alpha \in \{0.002, 0.02, 0.2\}$, which is what is predicted in (1). We can see perfect match for this prediction in for the incidence $t = 1$ and number of sets $n = 1$. However, this is not the case for $t > 1$ or $n > 1$. The same data were given to a linear regression model shown in Figure 5. The points in Figure 5 correspond to the same points in Figure 4 after being projected into the estimated linear model. The figure shows that there is useful pattern that we use to compute m for $t > 2, n > 2$. We discuss this topic further in Section 5.2, and why the points do not fall exactly on the linear regression line.

In Figure 6 we plot the multiplicative error obtained for $\alpha = 0.02$ and $\epsilon = 3$ for each setting and all t-incidences. Since three Bloom filters are released then the total privacy budget is 9. We can see that for certain settings the multiplicative error is somewhat close to α as desired while for others it is far. For instance the worst-performing setting is OTHERS. This is because this particular setting has very small values for the t-incidences as it was collected by researchers on their commute. However the best-performing settings are POLITICS1, POLITICS2, and THEMALL, are probably due to

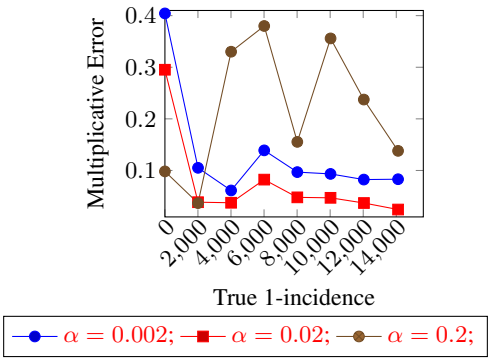

Fig. 2. Results for $t = 1$. The x axis is the true value being estimated (1-incidence) and the y axis is the multiplicative error on the estimate given by our algorithm. The data points were passed through an exponential moving average with weight parameter 0.3. Data points also cover most of the x axis but we plot only one point at steps of 2000, after the exponential moving average has taken place. We observe that the multiplicative error is acceptable for all considered values of α but is best at $\alpha = 0.02$.

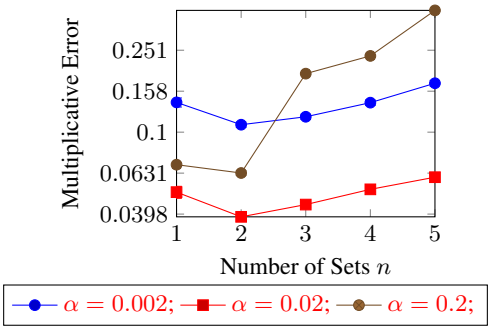

Fig. 3. Results for $t = 1$. The x axis represents the number n of sets considered and the y axis (log scale) is the multiplicative error on the estimate of the 1-incidence given by our algorithm.

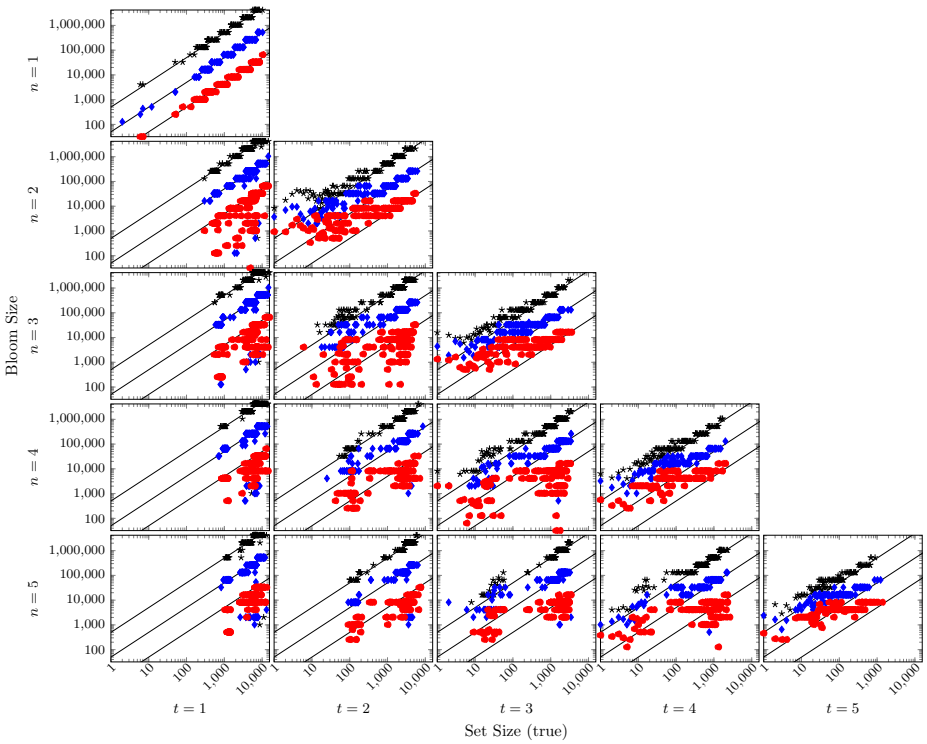

Fig. 4. Raw data. Raw lines are x/α, for $\alpha \in \{0.002, 0.02, 0.2\}$. The plain observed data are the shown points. Black is for $\alpha = 0.002$, blue is for $\alpha = 0.02$, and red is for $\alpha = 0.2$. $k = 1$.

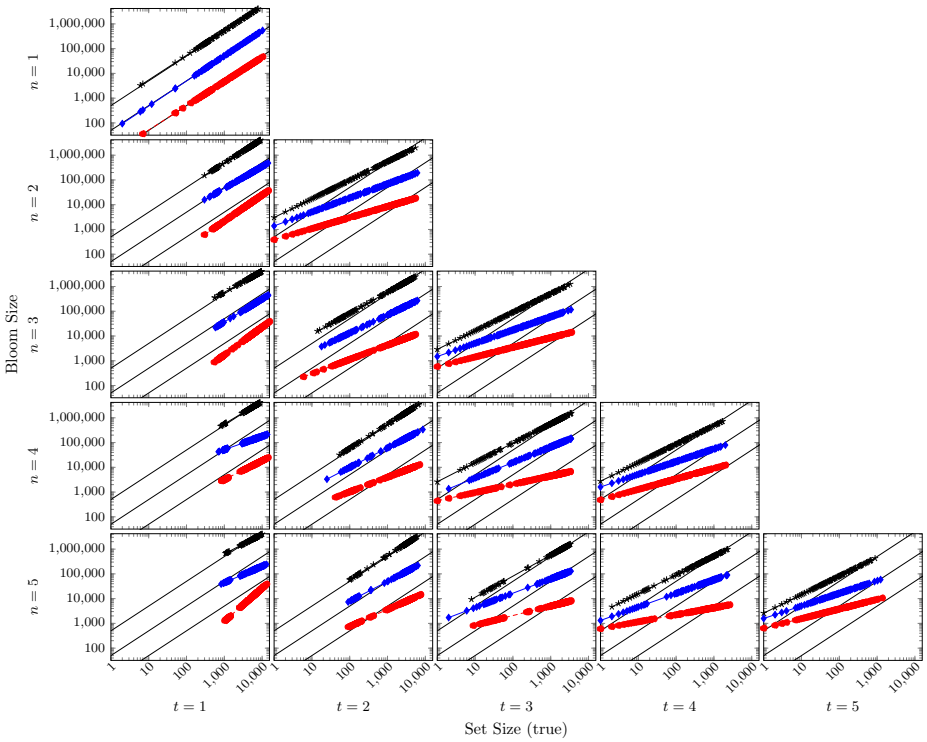

Fig. 5. Fitted data. Raw lines are x/α, for $\alpha \in \{0.002, 0.02, 0.2\}$. Other lines are linear regression. Black is for $\alpha = 0.002$, blue is for $\alpha = 0.02$, and red is for $\alpha = 0.2$. $k = 1$.

the fact that they were collected on a single day and thus exhibit more homogeneity and greater values for the t-incidences.

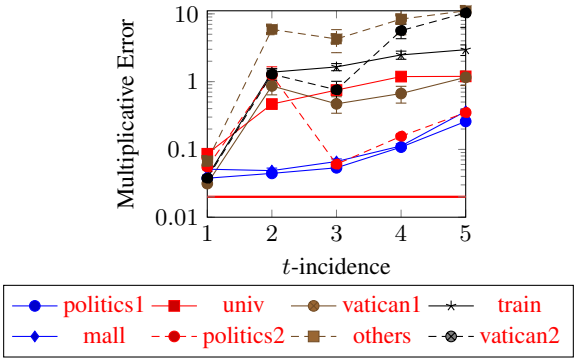

Fig. 6. Results for $\alpha = 0.02$. The x axis indicates the t of the incidence being estimated while the y axis (log scale) represents the multiplicative error on the estimate given by our algorithm. The flat red line indicates the desired multiplicative error $\alpha = 0.02$. The vertical bars represent 95% confidence interval around the mean.

5 Discussion

5.1 How to Know the Expected Size s

We need to know the expected set size s in advance to specify the size of the Bloom filter to use in order for its density to be close to the given α. Without knowing the set size in advance we may have to try a large number of values for m which would be bad for privacy or for utility. There are many ways to predict roughly the expected size s of MAC addresses. The simplest one is using historical data. However, when such data is not available we may look at the specific case at hand. If the data is to be captured at an event held within a building (a room or a stadium) we can know that s will never exceed the maximum holding capacity for this building. If the event is outside and is based on tickets like open space concerts or most planned events, the organizers know how many tickets have been sold and roughly how many people are going to show up. Finally, we remark that some of these approaches may be also applicable for the expected size of the t-incidences (such as historical data). We can know for instance that the intersection between two sets captured simultaneously by two access points that are geographically very far away is likely to be zero.

5.2 Select m, Given Expected t-incidence, Such That t-density is Close to α for $t > 1$

For $t = 1$ we know that for a set of cardinality s (*e.g.*, of 1-incidence s), its 1-density is

$$1 - (1 - 1/m)^s \quad . \tag{2}$$

Thus we can set the size m of a Bloom filter according to (1) to guarantee that the 1-density is close to α. In this section we are going to explore the case for $t = 2$ of two sets A and B, which should generalize to $t > 2$. We follow the same steps as (2). First we establish the 2-density as as a function of m and the expected 1-incidence and 2-incidence of A and B. Notice that the 1-incidence is $|A \Delta B|$ and the 2-incidence is $|A \cap B|$. It is clear thus that given those two expected values and this formula, we can select m to give a certain 2-density. Actually we will establish an inequality instead of the formula. Note that (2) is not an equation but an approximation with no clear bounds, so our results are stronger since the inequality establish a probabilistic bounds. One direction of the inequality that we will not mention in the subsequent paragraphs comes from the fact that the t-incidence is bounded from below by the t-density (which we denote in this section as d_t) times m. That is the t-incidence is $\geqslant m d_t$. In the following, we only discuss the other direction; bounding the t-incidence from above.

The equation (2) takes collisions into account. However the collision model for 2-density and higher or even for 1-density for more than one set is more complicated. To clarify the issue we will give an example. Consider the relatively simple case of $n = 2$ of two sets A and B, $n = 2$. Consider a hash table with chaining as collision resolution policy. We first insert all elements of A. When a bucket is touched by A, it is labeled RED. Then we insert elements of B in the following manner: If the bucket is empty or marked BLUE, insert the element and mark the bucket BLUE. If it was marked RED or BLACK, insert the element and mark it BLACK. Let X_i be the number of buckets of length i marked either RED or BLUE, and Z_i be the number of buckets of length i marked BLACK. Then we know that the 1-incidence is $\sum_{i>0} i X_i$ and that 1-density is $d_1 \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i>0} X_i$. We also know that the 2-incidence is $\sum_{i>0} i Z_i$, and the 2-density is $d_2 \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i>0} Z_i$. Therefore 1-incidence is the $m d_1 + \sum_{i \geqslant 2} (i-1) X_i$ and 2-incidence is $m d_2 + \sum_{i \geqslant 2} (i-1) Z_i$. Then let

Let $\overline{X}$ be the number of pairwise collisions in the 1-set $A \Delta B$, and $\overline{Z}$ be the number of pairwise collisions in the 2-set $A \cap B$. Then we know that $\overline{X} = \sum_{i \geqslant 2} \binom{i}{2} X_i = \sum_{i \geqslant 2} i(i-1) X_i \geqslant \sum_{i \geqslant 2} (i-1) X_i$ therefore the 1-incidence is bounded from above by $d_1 + \overline{X}$. Similar reasoning leads to that the 2-incidence is bounded from above by $d_2 + \overline{Z}$.

We will bound $\overline{X}$ and $\overline{Z}$ by some function of $A \setminus B, A \cap B, B \setminus A$. We know that $\overline{X}$ can come only from collisions either within $A \setminus B$ or within $B \setminus A$. From Lemmas 1 and 2 in [6] we know that the mean of $\overline{X}$ is less than $\left(|A \setminus B|^2 + |B \setminus A|^2\right)/m \leqslant 2|A \Delta B|^2/m$ and variance not greater than that.

However, $\overline{Z}$ is more versatile. Nonetheless, it has to involve elements from both sets. In particular it can come from these four types of collisions: 1) $A \cap B$ and itself, 2) $A \cap B$ and $A \setminus B$, 3) $A \cap B$ and $B \setminus A$, 4) $A \setminus B$ and $B \setminus A$. So the expectation for

$\overline{Y}$ is less than

$$\begin{aligned}
&\frac{1}{m}\big[|A\cap B|\big(|A\cap B|+|A\setminus B|+|B\setminus A|\big)+|A\setminus B||B\setminus A|\big]\\
&=\frac{1}{m}\big(|A\cap B|^2+|A\Delta B||A\cap B|+|A\setminus B||B\setminus A|\big)\\
&\leqslant\frac{1}{m}\big(|A\cap B|^2+|A\Delta B||A\cap B|+|A\Delta B|^2\big)\\
&\leqslant\frac{1}{m}\big(|A\cap B|^2+2|A\Delta B||A\cap B|+|A\Delta B|^2\big)\\
&=\frac{1}{m}\big(|A\cap B|+|A\Delta B|\big)^2\\
&=\frac{1}{m}|A\cup B|^2\ .
\end{aligned}$$

This is because it can be shown that for 4-independent hash function that the pairwise collisions between two sets of cardinalities a and b has mean ab/m and variance not exceeding that. So with similar logic we do the same as we did for the 1-incidence. Analyzing the collision behavior for $t > 2$ and subsequently selecting m for these cases may follow the same steps.

5.3 Alternative Way for Selecting the Bloom Filter Size m

Instead of anticipating a particular cardinality for the incoming set, we can set the Bloom size to a fixed value, regardless of application. As the set is being accumulated over time we may release the current Bloom filter as its estimated reaches density α. Upcoming items are then stored in a fresh Bloom filter, and so on. There are several advantages to this method. First, since the temporal dimension is not fixed all sets will have sizes close to each others, and it is unlikely to have a set which is exceptionally small or exceptionally big. For instance, during rush hours where the set would have been big, the data structure would adapt and release several Bloom filters, automatically providing higher granularity to rush hours. Moreover, users whose data are registered will know that a released Bloom filter will likely contain many people and not just a few one, even if they happen to pass by the Wi-Fi router at a non-busy time (like 4 in the morning). This also means that the privacy guarantees would not depend on the time of day. Additionally, since this way m can fixed beforehand, special hardware can be designed with this particular value of m in mind. This may be desirable in high traffic situations (not necessarily Wi-Fi related, like wire sniffing) in which hardware is needed to speedup data capture.

6 Conclusion and Future Work

We presented an algorithm to estimate, up to multiplicative error $O(\alpha)$, the t-incidence of n sets given three ϵ-differentially pan-private Bloom filters. The goal this algorithm was to provide a privacy-preserving set operations primitives to aid in designing and implementing physical mobility analytics. We considered an example of Wi-Fi mobility

analytics but our framework extends to any platform that can represent mobility data as sets of user/device unique IDs captured by stationary sensors. Experimental validation of our results for $t = 1$. Our results showed promising results for $t > 1$ in case the sets are dense. We also provided in depth discussion of how to extend it to the more challenging case of $t = 2$ was provided that sheds light as well into the $t > 2$ case. As future work we would like to consider providing a complete framework and analysis with provable error bounds for any $t \geqslant 1$.

References

1. Mohammad Alaggan, Mathieu Cunche, and Marine Minier. Frequency of Elements in Sanitized Streams and t-Incidence Estimation from Differentially Pan-Private Bloom Filters. Under submission, 2016.
2. Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. BLIP: Non-Interactive Differentially-Private Similarity Computation on Bloom Filters. In *Proceedings of the 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS12)*, Toronto, Canada, October, 2012. to appear.
3. Mohammad Alaggan, Sébastien Gambs, Stan Matwin, and Mohammed Tuhin. Sanitization of Call Detail Records via Differentially-Private Bloom Filters. In Pierangela Samarati, editor, *Data and Applications Security and Privacy XXIX - 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, volume 9149 of *Lecture Notes in Computer Science*, pages 223–230. Springer, 2015.
4. Marco V. Barbera, Alessandro Epasto, Alessandro Mei, Sokol Kosta, Vasile C. Perta, and Julinda Stefa. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Downloaded from http://crawdad.org/sapienza/probe-requests/20130910, September 2013.
5. Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM*, 13(7):422–426, July 1970.
6. Andrei Z. Broder and Anna R. Karlin. Multilevel adaptive hashing. In David S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California.*, pages 43–53. SIAM, 1990.
7. Siraj Datoo. How tracking customers in-store will soon be the norm, January 2014. [Online; posted 10-January-2014].
8. Levent Demir, Mathieu Cunche, and Cédric Lauradoux. Analysing the privacy policies of Wi-Fi trackers. In *Workshop on Physical Analytics*, Bretton Woods, United States, June 2014. ACM.
9. Cynthia Dwork. Differential Privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06), Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12, Venice, Italy, 2006. Springer.
10. Cynthia Dwork and Moni Naor. On the Difficulties of Disclosure Prevention in Statistical Databases or the Case for Differential Privacy. *Journal of Privacy and Confidentiality*, 2(1):8, 2008.
11. Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-Private Streaming Algorithms. In Andrew Chi-Chih Yao, editor, *Proceedings of the 1st Symposium on Innovations in Computer Science (ICS'10)*, pages 66–80, Tsinghua University, Beijing, China, 2010. Tsinghua University Press.
12. Dan Goodin. No, this isn't a scene from minority report. This trash can is stalking you. *Ars Technica*, 2013.

13. Ben Greenstein, Ramakrishna Gummadi, Jeffrey Pang, Mike Y. Chen, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Can Ferris Bueller still have his day off? protecting privacy in the wireless era. In *Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, pages 10:1–10:6, Berkeley, CA, USA, 2007. USENIX Association.
14. Michael Kamp, Christine Kopp, Michael Mock, Mario Boley, and Michael May. Privacy-preserving mobility monitoring using sketches of stationary sensor readings. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, volume 8190 of *Lecture Notes in Computer Science*, pages 370–386. Springer, 2013.
15. ABM Musa and Jakob Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.
16. Future of Privacy Forum. Mobile Location Analytics Code of Conduct, 2013. http://www.futureofprivacy.org/wp-content/uploads/10.22.13-FINAL-MLA-Code.pdf.
17. Latanya Sweeney. My phone at your service. *Federal Trade Commission*, 2014.