Loca: A Location-Oblivious Co-location Attack in Crowds

Roberto Pasqua, Matthieu Roy, Gilles Trédan

Loca: A Location-Oblivious Co-location Attack in Crowds*

Roberto Pasqua
LAAS-CNRS
Université de Toulouse, CNRS
Toulouse, France
pasqua@laas.fr

Matthieu Roy
LAAS-CNRS
Université de Toulouse, CNRS
Toulouse, France
roy@laas.fr

Gilles Tredan
LAAS-CNRS
Université de Toulouse, CNRS
Toulouse, France
tredan@laas.fr

ABSTRACT

Recent studies have introduced co-location attacks as a powerful way to extract social information from location traces. However, these attacks all rely by some means on the position of targeted users. This requires the attacker to be able to locate either the user or the sensors detecting the user. Implicitly, it also forbids the use of these attacks on devices whose location is unknown.

In this paper, we consider attack scenarios where the attacker has no position information on users and devices sensing users. Such attack scenarios typically fit Internet of Things use-cases, where low-end devices are scattered in an environment that is unknown to the attacker: the sole source of information is a set of timestamped user/sensor proximity logs.

To exploit proximity logs, we describe LOCA, a location-oblivious co-location attack. Our approach exploits location-oblivious logs in two steps: *i*) we exploit users' flows between sensors to construct a virtual map of the sensors, and *ii*) we conduct a co-location attack based on that virtual map. Our tests on both synthetic and real datasets match up to 90% of the targeted social network with a surprisingly low number of sensors. These results greatly extend the scope of such co-location attacks, and hopefully awareness about their threat.

ACM Classification Keywords

K.4.1. Computers and Society: Public Policy Issues: Privacy.

Author Keywords

Colocation Attack; Privacy; Crowd Sensing; Ubiquitous Systems

INTRODUCTION AND PROBLEM STATEMENT

The massive diffusion of GPS-equipped smart phones and the tremendous growth of the data market raised concerns about the privacy of users. The balance between the benefits in terms of service, and the drawbacks in terms of exposed private information is now better understood. To achieve this understanding, a decisive step was to assess the amount of private information that was enclosed in mobility traces of users. We know that the location of individuals can reveal much about their activities, lifestyle, job satisfaction [8] or even social contacts. As such, the recognition that mobility traces represent sensitive information has widened.

In this context, co-location attacks have demonstrated the danger of implicitly revealing social networks by exploiting the geographical proximity of located users and translating them in terms of "social proximity" [5]. The work carried out in [5] exploits a set of GPS-geo-tagged pictures. It divides Earth in a roughly 80km sided grid and correlates users' co-locations on this grid. Due to the huge time and space span of the dataset, accidental co-locations are extremely rare, and users being repeatedly co-located are often socially tied, as they explain: "two people have almost a 60% chance —nearly 5,000 times the baseline probability— of having a social tie on Flickr when they have five co-occurrences at a temporal range of a day in distinct cells".

Their work highlights the following principle: if users can be repeatedly located (either directly using a GPS device, or indirectly by revealing their proximity to a located device), an attacker can isolate recurring physical proximities between users, and derive localized social contacts, that are in turn correlated to general social interactions. Hence, to be effective, the attacker needs to have access to precise location information of equipment, be they user-carried or fixed.

The connection between location inference and social ties has later been weakened in [3] where the authors assume an attacker able to trilaterate mobile phone users. To that end, they assume the attacker knows the position of 37 Access Points (APs) in a Wifi mesh over a 130m $\times$ 250m campus region. They also assume the attacker is able to sniff some of the packets exchanged between the APs. This allows them to trilaterate mobile users based on APs positions and the sniffed RSSI values (used as a proxy for AP-user distance). Then, they exploit the obtained positions using a probabilistic framework to infer relationships among the targeted users. Finally, they swiftly exploit the specificities of the academic structure (*i.e.* students divided in classes) to improve the attack results using a community detection algorithm.

*This work is supported by CNRS PEPS CLUE-WD

Thanks to the awareness raised by the location-privacy research thread, location is now generally considered as sensitive information. The corresponding attack surface is shrinking as users are more aware of the threat and take actions to hide their location information. Yet, the expected massive deployment of small connected objects that are not located due to cost reasons gives attackers access to huge amount of data collected on users possibly without their consent. Consider the following two scenarios:

Sniffing Smart Dust
UHF RFID readers can read passive tags at a range of 1m [13] while cheap micro-controllers with Bluetooth Low Energy client capabilities and low power consumption cost around $30 (*e.g.* RFduino). An attacker can deploy many such sensors for a relatively low price, and this deployment can be done quickly as no positioning of the sensor is required. Such sensors have a very low power consumption and need only to be fitted with an RTC clock and some storage capacity to log the encountered MAC addresses.

Network-based Eavesdropper
An attacker spies on the network activity generated by users' smart phones reacting to the proximity of IoT objects such as iBeacons from Apple, Eddystone from Google, BLE tags, or reactive advertising platforms.

Consider for instance iBeacons scattered in the environment (say, a shop) that interact with users' smart phones to provide users with an enriched experience [10]. This is typically done by "binding" a tag to some digital content on the cloud. Therefore, these interactions typically involve a network fetch from the smart phone of the user (e.g., a web page, a price). The tuple (smart phone IP, tag url, time-stamp) can be monitored at various levels of the network infrastructure.

These scenarios show that a huge number of day-to-day activities of users, including users' colocations, are or will soon be available to potential attackers. Although such personal information on users does not include location data, the main threat of these scenarios is the impossibility for users to even notice the data collection, hence to take action against this collection, and the ubiquitous nature of such data collection campaigns.

In this paper, we go beyond location-based co-location attacks by providing a method to interpolate social interactions from a co-location attack performed on location-oblivious activity traces gathered in one of the scenarios exemplified above. Removing the dependence on the knowledge of location to conduct co-location attacks clearly extends their threat and scope. Indeed, this implies that cheap *IoT* objects represent a potential support of such attacks. Moreover, such an attack can be conducted without knowing the geographical layout of the targeted area, or even knowing where the targeted area is: it can be based on monitoring co-locations, independently of where they happen, provided that co-occurrences of users detected by a device represent a real co-location of users.

The principle of this attack is as follows: we assume we are able to collect the proximity of each user to a fixed set of K sensors for which we do not know any information apart from that they have unique identifiers. We use proximity data of users in two ways: first we exploit the recorded proximities of all users as a whole, and use them to construct a virtual map of the sensed space. Then we exploit "trajectories" of users in this virtual space to conduct a standard co-location attack. Through simulation on both real and synthetic datasets, we show that in dense enough environments, such a virtual cartography is accurate enough to reveal social contacts of users and preferences.

Contributions

This work shows that co-location attacks can be performed without any information on the location of users or sniffing devices. Our approach advances the state of the art on the following points:

1. We show that it is indeed possible to perform location-oblivious co-location attacks, putting emphasis on the possible risks involved by the deployment of cheap and ubiquitous objects, as advocated by the envisioned Internet of Things.

2. We provide an algorithm that implements our solution. The actual implementation is based on mysql and R scripts, making it generic and easily reusable for further improvements and adaptations.

3. We identify the key parameters influencing the attack accuracy, namely the number of sensors, their range, and the deployment pattern.

4. We evaluate our approach on different datasets, showing our approach is efficient on data captured using different technologies and on synthetic datasets. In order to fully evaluate LOCA, we provide a comparison of our approach with the best, to our knowledge, existing work.

The paper is structured as follows. Section 2 contains definitions of the model and describes the algorithm of LOCA. Section 3 reports experimental results of the inference attack. Section 4 highlights related work and compares it with our approach. We propose possible mitigation strategies to reduce the effect of the LOCA attack in Section 5.

MODEL AND ALGORITHMS

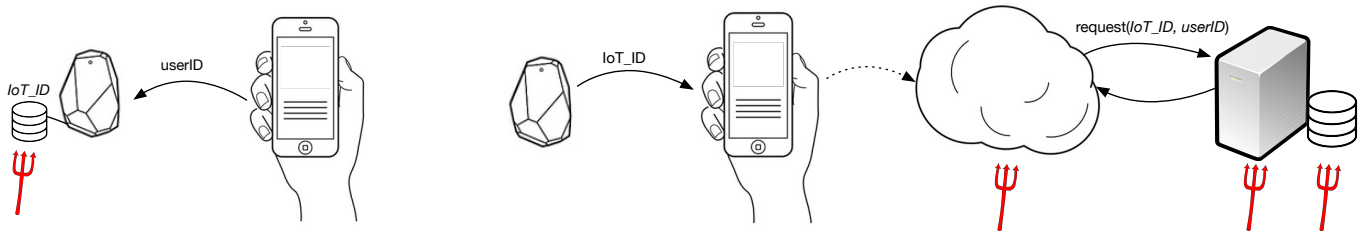
Attacker Model

We assume that an attacker has access to a set of *proximity logs*, captured by wireless equipped objects. More precisely, information available to the attacker has the following properties:

- Every user sensed by the objects has a unique identifier, e.g., a physical address or a cookie,

- Proximity logs contain timestamped detection of users' identifiers.

We do not assume a particular model for the attacker to get such information. As depicted in Figure 1, proximity logs may come from the objects themselves, if the attacker has physical access to IoT objects. It may also be eavesdropped in the network infrastructure, if the attacker controls a set of



(a) IoT device under control

(b) Compromised infrastructure or service provider

Figure 1: Possible attack surfaces represented by red arrows.

networks links. Additionally, proximity logs may come from the IoT service provider if the attacker is an insider —or if the service provider has been compromised.

From these logs, the attacker computes the number of users N, and the time span of the attack T, transforming proximity logs into a sequence snapshots for all time instants $t \in \{1,\ldots,T\}$. We consider that the attacker has K distinct proximity logs, coming from *immobile* sensors deployed in a given area. Each sensor is unaware of its position and is able to detect any subset of the N users that pass in its vicinity at each discrete instant $t \in \{1,\ldots,T\}$. Consequently, the total amount of information that is available is a set of K logs $(\mathsf{hist}_k)_{k=1..K}$ output by sensors, each one providing a set of detected users at every time instant $(\mathsf{hist}_k : [1,T] \to 2^{[1,N]})$.

Transformation Algorithms

In order to exploit low-level information hist_k, LOCA proceeds in three steps: 1) we compute aggregated proximity logs for each user, that provide the sequence of encountered sensors for each user during an experiment, 2) we construct a transition matrix that evaluates the flow of users between sensors during the experiment, and 3) we compute an interaction matrix that quantifies interactions between pairs of users, i.e., we generate the co-location events for each pair of users in the dataset.

Physical sensors: proximity logs

The following definitions formalize these steps, and the LOCA algorithm is described in Algorithm 1.

DEFINITION 1 (PROXIMITY LOG). *Given a set of histories* hist_k*, the proximity log* p_i *for user* i $(i \in 1\ldots N)$ *is defined as:*

$$p_i : \quad [1,T] \to [1,K]$$
$$t \quad \mapsto \begin{cases} k & if\ i \in \mathsf{hist}_k(t) \\ 0 & otherwise. \end{cases}$$

In this definition, we assume that sensors have non-overlapping range of detection. Such an assumption greatly simplifies presentation of later steps of the transformation algorithm.

Virtual sensors

Yet, in real systems, assuming that sensors shall not overlap is not realistic and too restrictive. As we will see later, our real attack algorithm does take into account sensors overlapping by means of *virtual sensors*. When two sensors overlap effectively, i.e. when two (or more) sensors detect the same user simultaneously, we assign this detection to a virtual sensor defined as the intersection of sensors that detected a user simultaneously. Notice that such a virtual sensors creation scheme is oblivious to the position of sensors, and is solely based on simultaneous detection of a user by different physical sensors.

Transition matrix: users flows

Starting from proximity logs of users, we can compute the *Transition Matrix R* that evaluates the transition of users between different sensors. Its elements $r_{k,k'}$ measure the flow of users that move from sensor k to sensor k'.

DEFINITION 2 (TRANSITION MATRIX). *For every* $(k,k') \in [1,K]^2$*, the element* $r_{k,k'}$ *of the Transition Matrix R is computed as follows:*

$$r_{k,k'} = \frac{1}{\sigma_k} \sum_{i\in[1,N]} |\{(i,t,t')\ s.t.\quad p_i(t)=k \wedge p_i(t')=k' \wedge p_i(t'')=0, \forall t'' \in]t,t'[\}|$$

$$with\ \sigma_k = \sum_{i\in[1,N],t\in[1,T]} \delta_{k,p_i(t)},$$

where δ *is the Kronecker delta.*

Due to physical constraints, the Transition Matrix is a non symmetric sparse matrix that contains an indirect estimation of real sensors distance, corresponding to "virtual distance".

Users' similarity score

To quantify interactions between two users and evaluate their co-location profile, we compute the similarity between their trajectories. We define the similarity index as

$$\forall (i,j) \in [1,N]^2, score(i,j) = \sum_{t\in[1,T]} r_{p_i(t),p_j(t)}. \qquad (1)$$

The definition above shows that similarity index is based on the virtual distance between sensors defined through the Transition Matrix.

Interaction matrix: co-locations

Finally, in order to study interactions between each pair of users, we define the *Interaction Matrix M*, a symmetric matrix that measures how two given users interact. This matrix is

Algorithm 1 LOCA algorithm to compute interaction matrix from sensors logs

Require: K sensor logs hist_k ▷ For brevity, the case of overlapping sensors/virtual sensors creation is omitted
Ensure: return the interaction matrix M

function COMPUTESIMILARITYMATRIX(hist_k)
  `int` $P[1\ldots N, 1\ldots T] = \{\{0\}\}$ ▷ Proximity logs for users, initially null
  `int` $R[1\ldots K, 1\ldots K] = \{\{0\}\}$ ▷ Transition matrix on sensors, initially null
  `int` $M[1\ldots N, 1\ldots N] = \{\{0\}\}$ ▷ Interaction matrix between users, initially null

  for $t = 1\ldots T$ **do** ▷ *First step*: compute proximity logs
    for $k = 1\ldots K$ **do**
      for all $i \in \mathsf{hist}_k(t)$ **do**
        $P[i,t] \leftarrow k$
      end for
    end for
  end for

  for $i = 1\ldots N$ **do** ▷ *Second step*: compute transitions
    for $t = 1\ldots T-1$ **do**
      if $P[i,t] \neq 0$ **then** ▷ User i is detected at time t:
        if $\{t' > t, P[i,t'] \neq 0\} \neq \emptyset$ **then** ▷ If user i is detected after t,
          let $t_i = \min\{t' > t, P[i,t'] \neq 0\}$
          $R[P[i,t], P[i,t_i]] \leftarrow R[P[i,t], P[i,t_i]] + 1$ ▷ then update transition matrix accordingly.
        end if
      end if
    end for
  end for
  for $k, k' = 1\ldots K$ **do**
    $R[k,k'] \leftarrow R[k,k'] / \sum_{t\in T} |\mathsf{hist}_k(t)|$ ▷ Normalize transitions
  end for

  for $i = 1\ldots N$ **do**
    for $j = i+1\ldots N$ **do**
      $score \leftarrow 0$
      for $t = 1\ldots T$ **do**
        $score \leftarrow score + R[P[i,t]], P[j,t]]$ ▷ *Third step*: compute interactions based on similarity
      end for
      $M[i,j] \leftarrow score$
      $M[j,i] \leftarrow score$
    end for
  end for
  return M ▷ Output interaction matrix
end function

the result of the co-location inference attack and contains the whole amount of co-location events detected during the experiment.

DEFINITION 3 (INTERACTION MATRIX). *Element $m_{i,j}$ of the Interaction Matrix M is defined as:*

$$m_{i,j} = m_{j,i} = \begin{cases} score(i,j) & \text{if } i \neq j. \\ 0 & \text{if } i = j. \end{cases} \quad (2)$$

Algorithm 1 describes the complete code used to compute the interaction matrix.

To sum up, LOCA exploits proximity logs in two complementary ways. First, to compute similarity between users, in a quite classical way. Second, to establish a virtual "map" between sensors: as we will see in the next Section, this map is an estimation of sensors distance that is sufficient to conduct an efficient colocation attack.

EXPERIMENTAL RESULTS

In this section, we present simulation results quantifying the accuracy and sensitivity of LOCA, as presented in Section 2.2.

Datasets

The LOCA algorithm has been tested on three real-world datasets and a set of synthetic ones, described hereafter. Each dataset consist in two parts: a) a set of users location traces and b) a ground truth representing for the real number of social interactions of each pair of users. The difficulty of acquiring datasets that contain both position information and social contacts severely constrained the number of datasets that could be exploited.

Synthetic
We first test the accuracy of the inference on a broad range of synthetic datasets generated thanks to the `pymobility` generator [1]. Table 1 contains the settings used to generate the datasets. We configure `pymobility` to simulate two types of gatherings, one corresponding to a *High Density* situation (hereafter, *HD*) with 40 individuals in a 100m^2 area, and one representing a *Low Density* situation (hereafter, *LD*) with again 40 individuals in a 900m^2 area. For both gatherings, we generate 10 independent mobility traces of 1000 seconds using Random Walk (rw), Random Waypoint (rwp), Random Direction (rd), Truncated Levy Walk (tlw), Gauss-Markov (gm), Reference Point Group Mobility model (rpgm), Time-variant Community (trw). For each of these traces, we also generate the corresponding contact traces using a distance-based interaction model as our ground truth, adapted from [12].

Real/SOUK
The SOUK data set contains precise trajectories of 45 users during a 45 minutes cocktail party [11], captured using ultra-wide band real-time location technology in a 100m^2 (10m $\times$ 10m) closed space. The ground truth consists of detected social interactions using the algorithm provided in [11]. Both dataset and algorithms can be freely downloaded [15].

Real/MILANO
The MILANO data set is similar to the SOUK dataset. It was captured using a similar technology and contains trajectories of 64 users during a 45 minutes cocktail party in a 225m^2 (15m $\times$ 15m) closed space. The ground truth consists of detected social interactions using the algorithm provided in [11, 15].

Real/Badges
The Badge dataset [7] contains trajectories of 39 employees equipped with active sociometric badges in an office environment during approximately one month. The considered ground truth consist of the cumulated face-to-face interaction time as recorded by the RFID badges. Even though in total 39 employees were equipped with badges, not all employees were present everyday. We only retained days where strictly more than 20 employees were present.

Metrics and Methods

For each dataset, we virtually "deployed" a set of K co-location sensors in the experimental space. The set of K proximity traces extracted from each sensor is then the only input of the inference algorithm: we exploit the sensor traces without relying at any point on their location. To measure the accuracy of the attack, we compare the results obtained from the attack represented by an interaction matrix M with the ground truth matrix G obtained from the original datasets.

	Space $[m]$		Nodes			Step
Name	Width	Length	N	v_{min}	V_{max}	T
HD	10	10	40	0.01 m/s	0.5 m/s	1000
LD	30	30	40	0.01 m/s	0.5 m/s	1000

Table 1: Input parameters.

The ground truth matrix is similar to the interaction matrix M in the sense that it contains the real interaction behavior for each pair of users.

To measure the accuracy of the inference, we rely on two metrics:

1. **Global Social Network Inference:** We measure the accuracy at which the attack identifies the strongest social ties of the ground truth social network G. To do so, we only target the K_{in} pairs of G having the highest number of interactions.

   We use Receiver Operating Characteristic (hereafter, ROC curves) analysis to compare different models by representing the evolution of the true positive rate (in our case: the predicted user pair is indeed one of the K_{in} strongest social ties) as a function of the false positive rate (in our case: the predicted user pair is not one of the K_{in} strongest ties).

   The Area Under the Receiver Operating Characteristic curve (or Area Under Curve, hereafter denoted as AUC) summarizes the obtained inference results. It can be interpreted as the probability of LOCA to assign a higher score to a randomly chosen user pair in K_{in} than to a randomly chosen user pair not in K_{in}. This methodology is commonly used in classification analysis to determine which of the studied models has the best prediction capabilities.

   We typically set $\mathsf{K}_{\mathsf{in}} = 3N$ to obtain a connected social network with an average degree equal to 3. Yet, due to the different levels of social activity of the nodes, we often observe a huge bias towards more social nodes.

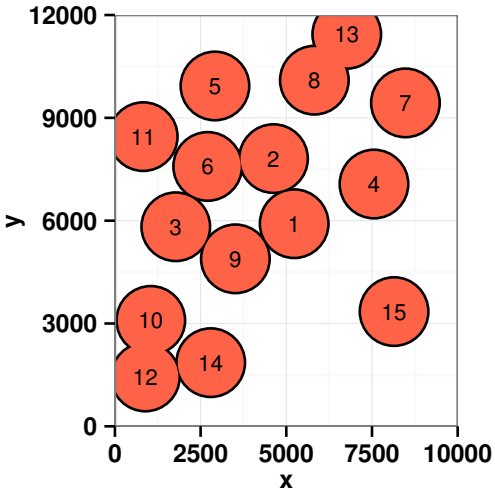
2. **Local Best Friend Inference:** For each node $i \in [1,N]$, we measure the size of the intersection between lists of i's 10 best friends (i.e. strongest social ties) in G and in M: we measure the recall@10 for each user. This is a tougher test for LOCA inference since we remove the bias toward more social individuals to also target weakly socialized users for which data is mechanically sparser.

To provide a comparison baseline, we also assessed the accuracy of the co-location attack presented in [3]. As this work is, to the best of our knowledge, the closest to our proposal to date, we implemented their approach to compare it against our algorithm. In the following, we refer to this approach as the State of the Art (SoA), obtained using the full trajectories. The gap between SoA and LOCA inference accuracy therefore measures the added cost of having no position information.

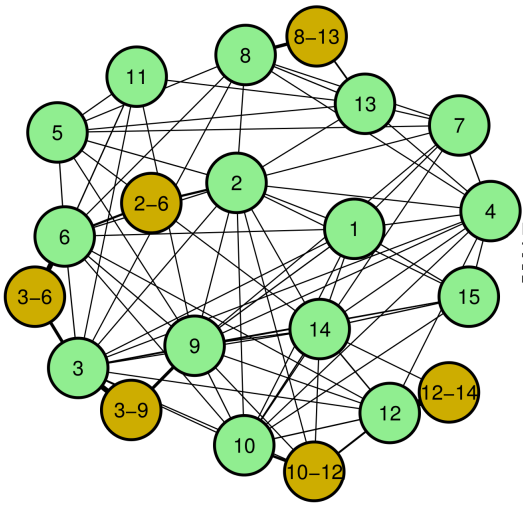
Sensors Deployment Strategies

Two parameters greatly impact the inference accuracy: the number of sensors K, and their range. These two parameters indirectly define the area covered by sensors. Intuitively, and as experiments will confirm, the higher the coverage achieved by sensors, the more accurate the attack is.

But another important parameter influencing the attack success is the density of users: sensors should be placed in dense areas in order to gather as many records as possible. Yet, adversary's control over this parameter is limited as he might have no control over sensors placement, or might not know a



(a) Physical sensors deployment: Density



(b) Computed virtual map —virtual sensors indicated in yellow are labelled with intersecting physical sensors label set.

Figure 2: (a) Deployment of 15 sensors in SOUK cocktail-room according to the Density strategy.
(b) Representation of the *Transition Matrix* as a weighted graph laid out using the Fruchterman and Reingold algorithm.

priori where users will densely regroup. We therefore study the following strategies:

- *Grid*: sensors are deployed along a regular grid starting from the center of the area.
- *Density*: sensors are placed using a priori knowledge of high-density areas (i.e. where people will gather and where social interactions are more likely to happen). This deployment represents the best possible situation and provides an upper bound of the sensor placement impact. Such information could have been obtained through observation of previous similar gatherings or exterior knowledge (e.g. metro platforms rather that metro corridors), or by targeting Points of Interest.
- *Spiral*: sensors are placed starting from the center of the area, following a counter-clockwise spiral extending up to the periphery of the area.
- *Random*: in this strategy, sensors are randomly located in the experimental area.

Note that the impact of poor sensor placement on LOCA is twofold: first, badly located sensors will provide very few co-locations, hindering the score computation, and second badly located sensors will degrade the relation between the virtual distances matrix of sensors (the Transition matrix) and the actual physical distances of sensors.

The last parameter conditioning the covered area is the sensors detecting radius. A 100 cm range could for instance represent the immediate range of an iBeacon device or a very conservative read range for an UHF RFID passive tag reader [13], whereas BLE range can be set by the programmer for up to 50 cm in practical contexts. The impact of range is detailed in Section 3.5.

Quality of the Virtual Mapping

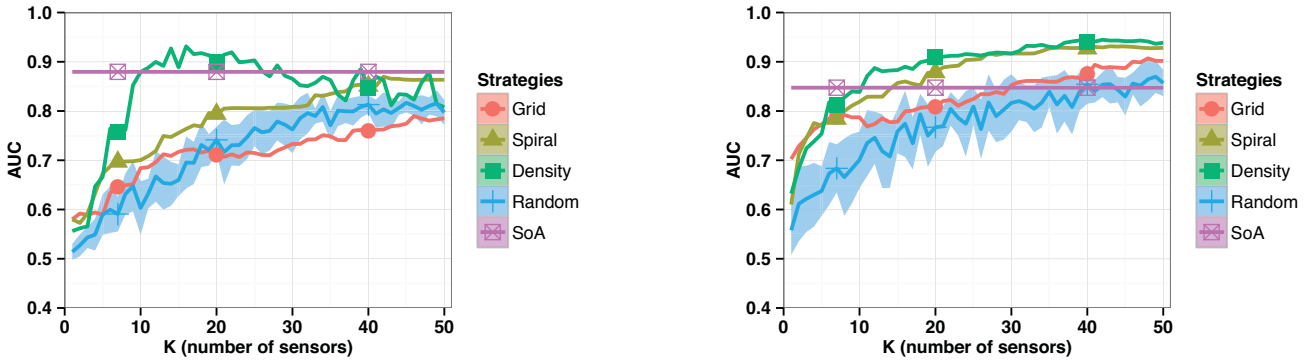
At the heart of our approach is the exploitation of users' sensor to sensor flows to estimate the distance between sensors through the construction of the Transition Matrix. Implicitly, we assume that the number of individuals going from one sensor to another is related to the distance between sensors.

Figure 2 illustrates the correlation between the real positions of sensors (Figure 2(a) – *left*) and the virtual positions obtained using the *Transition Matrix* values, represented using a Fruchterman-Reingold layout algorithm (Figure 2(b) – *right*). One can globally observe a good match: close sensors have high values in the transition matrix (symbolized by bold lines on the right representation), and therefore end up with a low "virtual distance" when computing the LOCA interaction matrix. Figure 2(b) depicts an example of virtual sensors as created by the LOCA algorithm in the case two sensors overlap (the same strategy is applied for overlapping of more than two sensors). The virtual sensors are named after the concatenation of identifiers of intersecting sensors (*e.g.*, $3-9$ corresponds to the overlapping of sensors 3 and 9).

Results

Synthetic/global

Table 2 summarizes the inference accuracy AUC for the different considered synthetic models. High density setups perform consistently better. This is explained by the relative area covered with sensors: the *HD* setup area is 9 times smaller than the *LD* area, resulting in a higher coverage with the same number of sensors. A high coverage provides a wealth of data to



(a) SOUK (b) MILANO

Figure 3: AUC results of the (a) SOUK and (b) MILANO datasets for different deployment strategies.

		Strategy			
Model	Config	Grid	Spiral	Density	Random
gm	*HD*	0.651	0.719	0.790	0.703
gm	*LD*	0.523	0.540	0.553	0.551
rd	*HD*	0.663	0.738	0.840	0.752
rd	*LD*	0.529	0.553	0.555	0.557
rpgm	*HD*	0.834	0.872	0.922	0.870
rpgm	*LD*	0.696	0.700	0.798	0.762
rw	*HD*	0.696	0.766	0.916	0.820
rw	*LD*	0.524	0.545	0.722	0.658
rwp	*HD*	0.685	0.735	0.785	0.698
rwp	*LD*	0.557	0.604	0.580	0.547
tlw	*HD*	0.665	0.723	0.850	0.753
tlw	*LD*	0.520	0.534	0.600	0.561
trw	*HD*	0.837	0.874	0.898	0.860
trw	*LD*	0.650	0.685	0.770	0.748

Table 2: Global network inference using LOCA on synthetic dataset: AUC of the inference accuracy using 15 sensors of 1 m range.

accurately estimate relative distance of sensors and ensures that most of the interactions are captured.

Not surprisingly, models where social interactions have an impact on the physical proximity of the users (namely *trw* and *rpgm*) are more accurately inferred than purely random ones, even in the worst case of a random deployment strategy.

Real/global

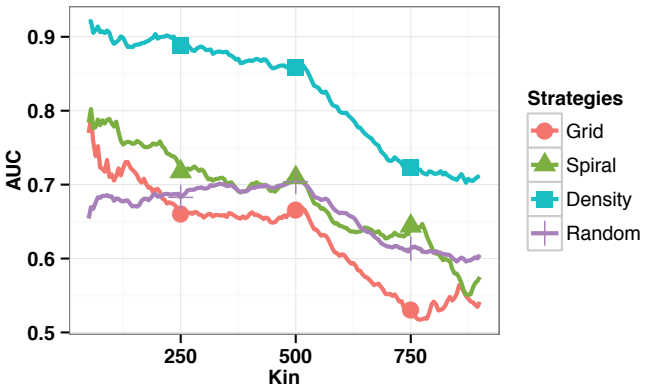
Figure 3 presents the AUC results for SOUK and MILANO datasets global inference using the different strategies and a varying number of sensors. It reads the following: for $K = 15$ sensors deployed during the SOUK cocktail, the AUC of LOCA inference is 66% for the Random strategy, 72% for the Grid strategy, 76% for the Spiral strategy and 89% for the Density strategy. Colored areas represent the standard deviation of the corresponding randomized strategies assessed over 10 independent runs. Since the SoA attack relies on full trajectories, it is not impacted by K, and yields 88% and 84% accuracies on SOUK and MILANO respectively.

On both datasets, the impact of K is clear: more sensors improve the inference accuracy. The only exception to this trend is the Density strategy on SOUK: it peaks quickly with surprisingly few sensors: 16 sensors (AUC is 93.14%) only cover about 12% of the 100 m^2 of the room, yet allow to identify the quasi-totality of the 150 most important social links. The slight decrease in the accuracy when more sensors are added is explained by the virtual sensor creation mechanism: as more sensors are only added in the densest areas, many sensing radii tend to overlap, leading to the creation of many "virtual sensors" that each contain little information. This globally degrades the quality of the Transition Matrix and therefore the quality of the inference.

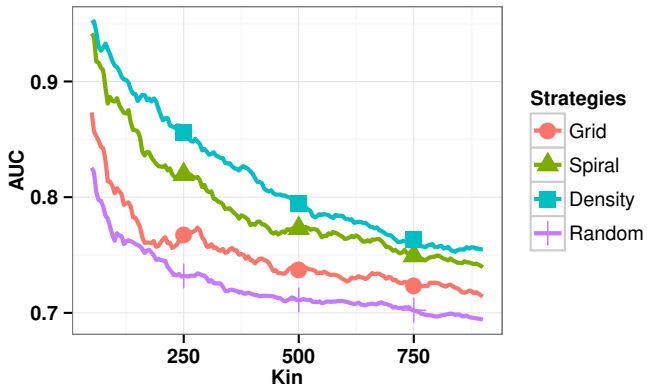
The Spiral strategy follows the Density results: this translates the fact that most of the social interactions happened in the center of the room on both datasets. Covering the SOUK area with 36 sensors (28% of the room covered) yields a 83% accuracy. To completely cover the SOUK area, 120 sensors are required for 89.2% accuracy (point not shown on picture), slightly less than Density strategy optimum. This again illustrates the fact that too many sensors add much noise to the process that can impede the detection with spurious proximities. The Random strategy is only efficient when many sensors are available, but yet yields accuracies close to the SoA on MILANO.

Figure 5 zooms on the accuracy results by presenting the ROC curves of the LOCA global network inference for $K = 15$ on the SOUK dataset. The Density strategy quickly identifies 50% of the 150 targeted links with a low false positive rate, similarly to the SoA approach. Both Grid and Random strategies provide the same detection performance with 12% false positives. The random strategy identifies some of the targeted links early on, but then performs no better than a purely random classifier.

Sensitivity analysis/global Figure 4 illustrates the sensitivity of LOCA to the targeted social tie strength. The left-hand side of the picture, SOUK, considers 45 participants, therefore the number of possible social links is almost 1000. In Figure 4(a),



(a) SOUK



(b) MILANO

Figure 4: Impact of K_{in} on SOUK (a) and MILANO (b) datasets, considering a deployment of 15 sensors with a 1 m range.

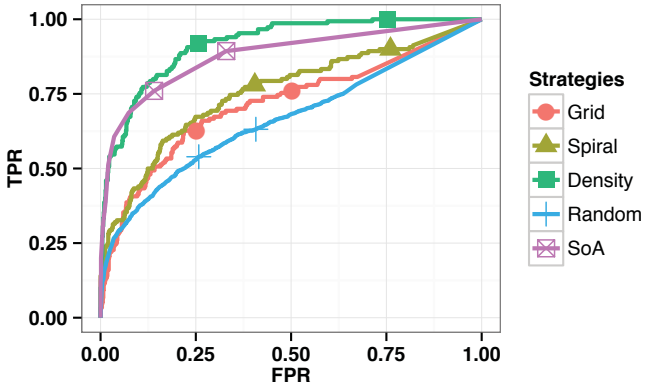


Figure 5: Detailed ROC for the SOUK dataset using 15 sensors: True Positive Rate (TPR) as a function of False Positive Rate (FRP).

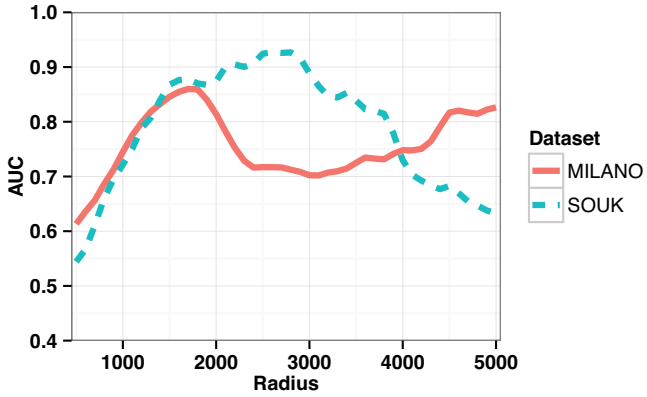


Figure 6: Impact of sensors range on inference accuracy for 15 sensors evenly distributed in space.

$K_{in} = 250$ corresponds to the detection of the 25% strongest social ties. Similarly, on the right-hand side of the figure, MILANO contains 64 users, resulting in slightly more than 2000 possible links. Hence, in Figure 4(b), $K_{in} = 250$ corresponds to the detection of the first eighth of social ties.

As K_{in} grows, we measure the ability of LOCA to detect weaker social ties. As stronger ties are easier to detect than weaker ones, the global inference accuracy decreases when K_{in} grows. The hierarchy of strategies (*e.g.* density is more efficient than random) is globally unchanged, whatever K_{in}.

Figure 6 evaluates the impact of range on detection accuracy. Due to the different sizes of SOUK and MILANO experimental spaces, the peak accuracy is achieved at different ranges: 2.6 m for SOUK and 1.7 m for MILANO. One can observe that 3 m range sensors perform poorly on the MILANO dataset.

This fact is partly explained by the combined effect of our virtual sensors creation scheme and the different density of users. In the SOUK experiment, the spatial density is higher than in the MILANO experiment, reducing the coverage of the assumption that co-locations and interactions are correlated. From this observation, one may intuitively assume the optimal range would be smaller for SOUK than for MILANO. Let us now consider our virtual sensor definition strategy: if (at least) two sensors overlap and (at least) one co-location happens in an intersection, then we create a virtual sensor whose sensing area is the intersection of the overlapping sensors. If sensors are densely deployed, many ranges will overlap, and therefore many virtual sensors will be created. If too many of these virtual sensors are created, each will end up with a little fraction of the (finite) localisation events, leading to a relatively poor distance estimation. Hence, the optimal range for a scattered social event is finally lower than the optimal range we measured in a dense social event.

Badges/global

Table 3 presents the AUC results for the Badges dataset. In this dataset, due to the specificities of the office environment,

Day:	4	5	6	7	8	11
Population:	24	28	31	32	29	23
AUC:	0.776	0.740	0.683	0.665	0.620	0.588

Table 3: Badges Dataset results.

we chose to place sensors near points of interests only: coffee and meeting rooms, printers, and managers' offices. In total, 30 sensors of 50 cm range are virtually deployed, covering approximately 7.5% of the office space. Results present a huge variability although each day is taken as an independent experiment. The low mobility of the users combined with the low number of social interactions happening per day could explain the lower accuracy of days 8 and 11.

SOUK and MILANO/local

Figure 7 offers a local perspective on LOCA accuracy. It represents for each Dataset the probability of having x of each individual's 10 best friends correctly inferred. Surprisingly, SOUK outperforms MILANO from this local perspective. Note that this does not contradict the global results as the objectives are rather different: detecting the strongest social ties among a whole network is different than detecting each participant's strongest social ties. The difference between SOUK and MILANO can be explained by the participants behaviors: in SOUK, most of the 45 participants stayed for the whole experimentation, whereas many of the 64 participants of the MILANO cocktail left quickly, providing LOCA with very few data to infer their local ties.

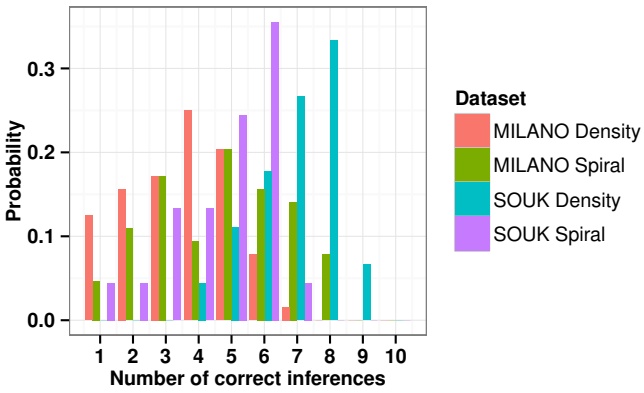


Figure 7: Probability to have exactly x out of 10 best friends correctly identified by LOCA using 15 sensors of 1 m range.

Together, these experimental results show the relatively good accuracy of LOCA inference considering the sparse information sources on which it relies. Moreover, they allow us to identify the key factors driving the inference accuracy: the number of sensors and their range, and above all the sensors deployment strategy. Indeed, having even a slight hint about the likely locus of future interactions (e.g. the Spiral strategy) considerably improves the inference accuracy over a Random sensor placement. Surprisingly enough, LOCA not only allows us to derive an accurate picture of the social network's most salient contacts, but also provides a fairly precise estimation of each users' individual contacts, as it correctly infers on average more than 6.5 out of the 10 strongest ties on the SOUK dataset for the Density strategy.

RELATED WORKS

The thread of co-location research started recently along with our ability to explore rich datasets containing both location and social information. One paradigmatic such dataset is the Reality Mining dataset and its analysis in [8]. Similarly, [5] exploited Flickr geo-tagged pictures at the global scale to infer Flickr declared friendship network, and [4] relied on the exploitation of both Location Based Systems and cell phone CDRs. Those works more generally question the interplay between physical proximity and social proximity, and typically rely on data having large geographical and temporal scales. Our work focuses on local observation of users' behaviors (as opposed to, say, day and country-wide CDR-based user tracking) in order to stick to the IoT-based sensor use case.

More recent studies exploit the connection of proximity and social ties with a privacy angle [3, 14]. These landmark works evaluate the privacy implications of wireless communications in office environments. Having equipped for 3 months a set of 80 university users with tracked mobile phones, the authors use the campus Wifi map to locate users and infer, from an eavesdropper perspective, the social contacts of the users as captured by the tracked mobile phone. Their attack assumes the adversary is able to "trilaterate" mobile phones by using both Access Point positions and exfiltered RSSI values. In [14], the authors study the impact of obfuscated location information on co-location attack accuracy. Our approach is both more localized in time and space, and assumes that no location information is available at all.

A closely related study [6] exploits another wireless-related information dataset to infer social contacts: the list of known SSIDs often broadcasted by smart phones to speedup the link establishment to APs. The authors exploit the similarity of SSID lists to infer social proximity. In this approach, the vast variety of encountered SSIDs allows them to efficiently fingerprint users. In contrast, our approach is generic as it does not depend on the specificities of one wireless protocol, and exploits only proximities.

MITIGATION STRATEGIES

We here present some suggestions to mitigate the success of LOCA-like attacks. Those strategies are known: in that respect LOCA simply extends the scope of co-location attacks against which strategies have already been elaborated.

A first step is to prevent LOCA from following the same individual when he/she moves across different sensors. This can be achieved for instance by mix-zones [2], by extending the scope of traditional Location Privacy Protection Mechanisms [16, 9], or by simply relying on the use of many pseudonyms —this last solution applies only if the traceability of a user across different sensors is not an important feature for the provided services.

Another mitigation is to limit the attack surface of such systems, by avoiding network operations, especially if they contain identifying information. One strategy is to design "silent protocols" such that the carried device cannot be inadvertently detected by an eavesdropping device. Since RFIDs can be passively read at distance, the users can store them in shielded wallets that are available on the market.

CONCLUSION

In this paper, we presented LOCA, a simple co-location inference algorithm that has the particularity of not relying on positioning information. Instead, it uses sensed data to estimate the physical distance between sensors. The so-constructed *virtual distances* are then used to conduct the co-location inference attack. Experimental results show that LOCA inference remains accurate even if the monitored space is sparsely covered, and highlights the importance of the sensor placement strategy in such contexts.

LOCA shows that location is not required to conduct co-location attacks. This conclusion strikes the potential privacy issue of trending IoT devices that could become potential attack surfaces for co-location attacks. Such attacks could be conducted either by directly monitoring users' proximity (i.e., altering the device's firmware to report nearby users), or for instance by monitoring the web page requests generated by close users' smart phones as LOCA requires no information about the sensors themselves.

Going further users' privacy, LOCA demonstrates a potential risk for service providers in the Internet of Things ecosystem: even though service providers may not store positioning information of deployed communicating objects, performing LOCA on the logs of users-objects interaction will provide lots of information on objects physical repartition. As such, by raising this issue over exploitation of users-objects interaction logs, we advocate the deployment of more decentralized architectures for sustainable ubiquitous systems deployment.

REFERENCES

1. André Panisson. 2015. `pymobility`: python implementation of mobility models. `https://github.com/panisson/pymobility`. (2015).
2. Alastair R Beresford and Frank Stajano. 2003. Location privacy in pervasive computing. *IEEE Pervasive computing* 1 (2003), 46–55.
3. Igor Bilogrevic, Kévin Huguenin, Murtuza Jadliwala, Florent Lopez, Jean-Pierre Hubaux, Philip Ginzboorg, and Valtteri Niemi. 2013. Inferring social ties in academic networks using short-range wireless communications. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 179–188.
4. Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1082–1090.
5. David J Crandall, Lars Backstrom, Dan Cosley, Siddharth Suri, Daniel Huttenlocher, and Jon Kleinberg. 2010. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences* 107, 52 (2010), 22436–22441.
6. Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. 2014. Linking wireless devices using information contained in Wi-Fi probe requests. *Pervasive and Mobile Computing* 11 (2014), 56–69.
7. Wen Dong, Daniel Olguin-Olguin, Benjamin Waber, Taemie Kim, and Alex Sandy Pentland. 2012. Mapping Organizational Dynamics with Body Sensor Networks. In *International Workshop on Wearable and Implantable Body Sensor Networks*, Vol. 0. IEEE Computer Society, Los Alamitos, CA, USA, 130–135.
8. Nathan Eagle, Alex Sandy Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences* 106, 36 (2009), 15274–15278.
9. Sheng Gao, Jianfeng Ma, Weisong Shi, and Guoxing Zhan. 2015. LTPPM: a location and trajectory privacy protection mechanism in participatory sensing. *Wireless Communications and Mobile Computing* 15, 1 (2015), 155–169.
10. Estimote Inc. 2016. Estimote Beacons. `http://www.estimote.com`. (2016).
11. Marc-Olivier Killijian, Matthieu Roy, Gilles Trédan, and Christophe Zanon. 2013. Souk: social observation of human kinetics. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 193–196.
12. Byoungjip Kim, Jin-Young Ha, SangJeong Lee, Seungwoo Kang, Youngki Lee, Yunseok Rhee, Lama Nachman, and Junehwa Song. 2011. Adnext: a visit-pattern-aware mobile advertising system for urban commercial complexes. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. ACM, 7–12.
13. Pavel V Nikitin and KVS Rao. 2006. Performance limitations of passive UHF RFID systems. In *IEEE Antennas and Propagation Society International Symposium*, Vol. 1011.
14. Alexandra-Mihaela Olteanu, Kévin Huguenin, Reza Shokri, and Jean-Pierre Hubaux. 2014. Quantifying the effect of co-location information on location privacy. In *Privacy Enhancing Technologies*. Springer, 184–203.
15. Matthieu Roy, Gilles Tredan, and Christophe Zanon. 2016. SOUK: Social & Spatial Observation of hUman Kinetics. `http://projects.laas.fr/souk/software`. (2016).
16. Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011. Quantifying location privacy. In *2011 IEEE Symposium on Security and Privacy*. IEEE, 247–262.