Non-Invasive User Tracking via Passive Sensing: Privacy Risks of Time-Series Occupancy Measurement

Xiao Wang
Carnegie Mellon University
xiaowang@cmu.edu

Patrick Tague
Carnegie Mellon University
tague@cmu.edu

ABSTRACT

A large-scale sensing infrastructure can collect ample data to benefit many real-world applications. One promising application scenario is building management. However, exposure of the sensor data potentially reveals private details about building users. In this paper, we investigate indoor location privacy as a motivating example to manifest potential privacy risks in smart buildings. We apply inference techniques to reconstruct users' location traces from room-level occupancy data. Unlike other types of surveillance that are dedicated to explicit tracking such as security cameras, time-series occupancy traces, as aggregated environmental measurements, are typically deemed privacy-preserving. Unfortunately, it may still reveal some of the same sensitive information as privacy-invasive sensing such as video surveillance. We conduct experiments using a publicly available dataset and synthetic data. Our results demonstrate the underlying privacy leakage via occupancy data. We further show how our evaluation can enable adaptive privacy mechanisms to control the information leakage by the sensing system.

Categories and Subject Descriptors

K.4.1 [**Public Policy Issues**]: Privacy; I [**Computing Methodologies**]: Artificial Intelligence—*inference engines, parameter learning, dynamic programming, heuristic methods*

Keywords

Location Privacy; Occupancy; Time-series; Utility-privacy Tradeoff; Stochastic Modelling

1. INTRODUCTION

With the advance in sensing technologies and the decrease in cost, power and size of solid-state devices, it is now possible to support ubiquitous sensing and actuation at scale with increasing temporal resolution [1]. During consistent sensing, ample data is collected, stored, and communicated which brings opportunities for a variety of data-driven services and applications [2–5]. Building management, among all kinds of application scenarios, benefits extensively from such a sensing infrastructure and inspires a variety of useful services regarding heating, cooling, illumination, safety and many other purposes. Residing on top of the sensing layer, service providers, either in-building or cloud-based, can be aware of building status and conditions through sensor measurements, and react properly and automatically to environmental variations, essentially creating a *smart building* [6–9].

Enabling this level of intelligence, however, inevitably exposes rich information about the building environment as well as its occupants because many types of sensor measurements are either subtly related to user behaviors or even user-centric in nature. Hence, the very process of generating, storing and transporting sensor data for building management presents inherent risk to the *privacy* of building users. Understanding these risks, quantifying the associated tradeoffs between privacy and utility, and designing smart building systems that respect users' privacy concerns while still meeting operational goals are all open problems.

To highlight the value of building sensory data and the possibility to exploit it for inferring private information, we consider as a motivating example the *occupancy* data, namely the number of users in each room. Given its value in building management, occupancy data has become one of the most sought-after pieces of analytical output of the sensing infrastructure, and it has been inspiring extensive research [10–14] as well as a number of commercial products. Many passive sensing or indoor tracking platforms can create real-time occupancy data with high accuracy [15–17].

Among a variety of applications, occupancy is highly valuable especially in heating, ventilation, and air conditioning (HVAC) systems [18–21]. There has been a considerable effort in industry and academia toward optimizing the HVAC system to enable intelligent controls in response to occupancy variations. While enjoying the benefits brought by occupancy data, people fail to realize potential risks in the data. Occupancy detection is not designed for surveillance purpose. It has been believed that occupancy data is privacy-preserving because it reports only the number of users and reveals identity of nobody [22]. This carefree attitude is risky. In this paper, we show that even aggregated measurements that appear privacy-preserving or privacy-enhancing are still subject to inference attacks using side information that can reveal some of the same sensitive information as privacy-invasive techniques such as video surveillance. With access to occupancy data over a period of time, a malicious or curious individual can possibly infer occupants' indoor locations since dynamics of occupancy leak information about mobility patterns, given the intuition that changes in room occupancy correlate to user transitions between rooms.

To understand this location privacy threat, we first study the adversarial aspect of the problem setting by developing inference tech-

AISec'14, November 7, 2014, Scottsdale, Arizona, USA.
Copyright 2014 ACM http://dx.doi.org/10.1145/2666652.2666655.

niques and algorithms to reconstruct user location traces from the time-series occupancy measurements being generated. Specifically, we build a stochastic framework to infer location traces and evaluate the inference performance. The framework is based on the *factorial hidden Markov model* [23]. The model captures the dependence of occupancy measurements on user locations and the Markovian property of user location transitions between rooms over time. Prior work has studied privacy issues in other types of sensor data, including smart meter readings [24, 25]. Unlike these heuristic or information theoretic approaches, we perform an *empirical and quantitative* study. We design the location inference attack and evaluate it quantitatively with real data. It can therefore explain how privacy breach possibilities can present realistic and achievable risks, and how we can quantify the risks.

Our quantitative treatment of the privacy problem also allows us to consider design of privacy-enhancing sensing systems, instead of only considering the adversarial angle. The performance of the HVAC system relies on the granularity of the occupancy data; with a high sensing rate, the infrastructure can continually sense the environment and enable prompt response to occupancy variations. In other words, the system enjoys more data utility. However, more fine-grained occupancy data, on the other hand, leads to more information leakage to the adversaries, i.e. reduced location privacy. This conflict essentially presents a tradeoff between *data utility and user privacy*. Therefore, knowing the variations of utility and privacy with respect to the system configurations, we can configure the sensing system to act in a privacy-conscious manner and make informed and automated decisions to achieve the optimal tradeoff, for example by minimizing privacy risk while satisfying a minimum requirement on sensing quality.

In this paper, leveraging previous research outcomes on location privacy and mature machine learning techniques, we present a stochastic framework for inferring locations of building users from the occupancy data. We achieve the following objectives.

- We demonstrate potential location privacy leakage from the occupancy data through a formal approach that allows us to reconstruct location traces of occupants.
- We show how our framework enables quantitative analysis of the inference performance and location privacy of users with respect to several impacting factors and design parameters of the sensing system.
- We illustrate how our results can potentially cast light on the design of privacy-aware sensing and actuation.

Our investigation of location privacy in smart buildings begins with a summary of related work. We then present our location inference framework, followed by a detailed description of the enabling machine learning techniques. We then provide detailed exposition of our proposed location inference attacks and experimental study. Finally, we show how our results can be used for adaptive privacy control by the sensing system.

2. RELATED WORK

Previous research has addressed the issue of explicitly tracking users or computing building occupancy as an intermediate statistic for more abstract building information collection. As examples, Manzoor et al. [15] proposed the use of RFID readers in doorways and passages to localize users. Hnat et al. developed the Doorjamb system [17] to achieve unobtrusive room-level tracking using doorway sensors. In our work, we show that it is able to track users using occupancy data instead of explicitly tracking with dedicated systems, which can be exploited by adversaries who have access to the occupancy data.

Occupancy measurements are useful to HVAC and many other systems. Occupancy detection has received much attention in recent years. To obtain occupancy data, previous work proposed the use of infrared sensors, acoustic sensors, CO_2 sensors, motion sensors, cameras and other types of sensors in couple with estimation theory, vision algorithms and machine learning techniques [10, 12, 13, 21]. Another straight-forward yet effective approach is to aggregate location data from indoor tracking system to generate occupancy outputs.

While offering rich utility for building management, the dedicated systems described above together with other general-purpose sensing systems open the breach of user privacy. McDaniel et al. [24] studied privacy issues in smart grid and revealed how attackers, heuristically, can infer user activities/behaviors through smart meter readings. They discussed the problem qualitatively and raised the concern about privacy risks in smart environments. Our work investigates specifically the location privacy issue via occupancy data and, in contrast, provides analytical results in addition to high level intuitions. Gruteser et al. [22] propose privacy-aware techniques to collect occupancy and location data without compromising the privacy of users; unfortunately, these techniques do not protect against our location inference approach. User privacy preservation in pervasive sensor-rich environments has been studied recently by Pallapa et al. [26,27] and by Hengartner [28], where the authors provided context-aware privacy preservation techniques that attempt to minimize privacy risks. Other studies of privacy in sensor-rich environments primarily study either location privacy or user/data anonymization [29–39]. Anonymization of the occupancy data, however, offers no protection against attackers who have direct access to the sensing database. Additionally, anonymization is not desired as management operations are usually room-specific and require room identity in nature.

There are many related studies on location privacy in particular. Gruteser and Grunwald [40] surveyed privacy issues related to location-based services, and introduced a quadtree-based algorithm to guarantee k-anonymity. Krumm proposed several location privacy protection schemes [41]. To evaluate the schemes, researchers proposed two popular metrics: k-anonymity and entropy-based criteria [42]. Shokri et al. showed that the two metrics are not adequate. They developed a probabilistic framework to quantify location privacy and further to analyze different protection strategies [43]. Their approach reconstructed outdoor location traces from anonymized and obfuscated traces. In our indoor scenario, we leverage the occupancy data instead of sanitized user locations.

A common goal for system design is to achieve satisfactory utility without sacrificing user privacy. Rajagopalan et al. [25] studied the tradeoff between utility and privacy in smart grid. To quantify this tradeoff, they applied information theoretic methods. We take a quantitative treatment as well, yet through a realizable approach. In contrast, the information theoretic bound might not be achievable by feasible algorithms. Eney at al. [44] proposed a theoretic scheme for balancing utility and privacy in smart sensor applications, in which the raw data is transformed. In our work, however, the adversaries have access to the raw occupancy data.

3. LOCATION INFERENCE FRAMEWORK

In this section, we present our framework for inferring indoor locations from time-series occupancy data. The framework specifies essential components under our problem settings. As illustrated in Figure 1, the framework comprises the use of observable occupancy data in conjunction with available context information

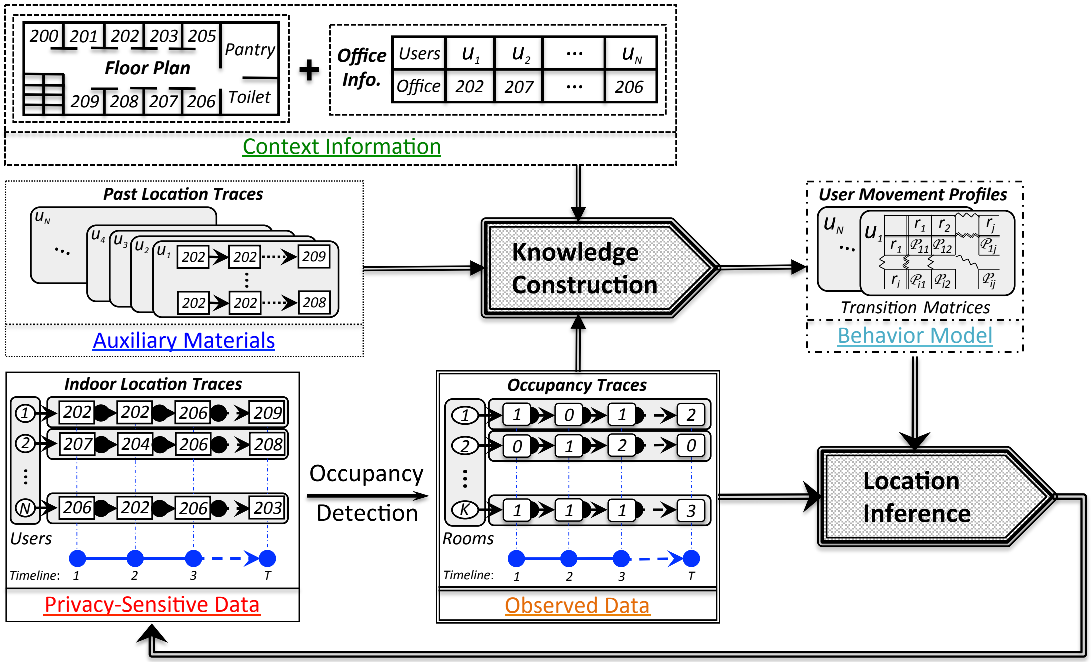

Figure 1: We illustrate the framework for indoor location inference from occupancy data. The privacy-sensitive location traces of building users correlate to the occupancy traces of rooms. The occupancy traces together with past location traces and available context information are fed to the knowledge construction module to create movement profiles for all users. The profile specifies a user's pattern with respect to transitioning between rooms. Using the profiles, the location inference module reconstructs location traces from the occupancy data during a certain period of time.

and past user location traces to construct the user movement profiles that constitute the knowledge of adversaries. The knowledge is fed to the inference module to enable reconstructing underlying user location traces from occupancy. The overall building management system includes a sensing infrastructure, a data repository and many data consumers (service providers). The malicious or curious individual can be the database administrator or anyone who has direct access to the sensor data, and thus all raw time-series occupancy data for all rooms.

In the following subsections, we formally define these components and describe the relations among them. We also propose metrics to evaluate our inference results. For the ease of presentation, we denote random variables using calligraphic letters, realizations of random variables using lower case letters, and sets from which the random variables take values using blackboard bold letters. For instance, random variable $\mathcal{X}$ takes value x from set $\mathbb{X}$.

3.1 Occupancy Trace

The adversary aims to infer user locations from room occupancy. Occupancy of a room is defined as the number of users in that room. Formally, the building area of interest consists of K rooms/regions. A special location is `away` which means users are not present in any of the rooms of interest. Defining the `away` location allows the adversary to partition a huge building area into several subareas (floors, sections, departments). The `away` state aids in defining user's transitions between subareas. As so, the adversary can perform a divide-and-conquer approach in which location inference is conducted in each subarea separately. We will revisit this point later on. Readers can refer to the floor plan in the Augsburg benchmark as an example [45]. The set of locations is denoted as $\mathbb{R} = \{r_1, r_2, \cdots, r_K\}$. The occupancy for room r_k at time t is a random variable denoted as $\mathcal{O}_t^{(k)}$ ($k = 1, 2, \cdots, K$). The occupancy variable takes a value from the set $\mathbb{O}_t^{(k)} = \{0, 1, \cdots, N\}$, where N is the total number of users in the building. We denote the occupancy for the building at time t as $\mathcal{O}_t = (\mathcal{O}_t^{(1)}, \mathcal{O}_t^{(2)}, \cdots, \mathcal{O}_t^{(K)})$. It takes values from a subset of $\mathbb{O}_t = \mathbb{O}_t^{(1)} \times \mathbb{O}_t^{(2)} \times \cdots \times \mathbb{O}_t^{(K)}$ subject to the constraint $\sum_{k=1}^{K} \mathcal{O}_t^{(k)} = N$.

The occupancy trace for each room is a timestamped sequence of occupancy measurements. Further, an occupancy trace for the tracking area is a time series of occupancy vectors for all rooms. The corresponding set of timestamps is an ordered set denoted as $\mathbb{T}_o = \{t_1, t_2, \cdots, t_T\}$. Those time instants are when there are occupancy changes. Since the detection system generates occupancy measurements periodically based on a sensing interval, timestamps in $\mathbb{T}_o$ are aligned to fixed sensing instants. Therefore, delay is expected in the timestamps compared to the instants when occupancy variations actually happen. We will emphasize this difference again when defining the location trace. The timestamp sequence also gives the length of the occupancy trace, i.e. T. For brevity, we denote the time sequence using only indices so that $\mathbb{T}_o = \{1, 2, \cdots, T\}$. The occupancy trace for the building area is thus a random process $\mathcal{O} = \{\mathcal{O}_t : t \in \mathbb{T}_o\}$.

3.2 Location Trace

We consider a total of N building users. The user set is denoted as $\mathbb{U} = \{u_1, u_2, \cdots, u_N\}$. The set $\mathbb{R}$ defines possible locations for all users. The location for user u_n at time t is a random variable denoted as $\mathcal{L}_t^{(n)}$ ($n = 1, 2, \cdots, N$), which takes values from location set $\mathbb{L}^{(n)} = \mathbb{R}$.

A location trace for a user is defined as a time series of locations visited by that user. For the purpose of evaluation, we have

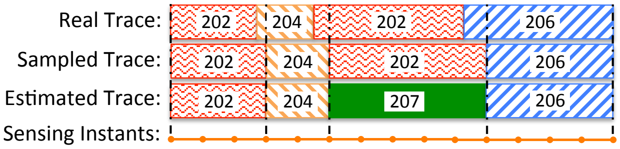

Figure 2: To distinguish between transition-based accuracy and time-averaged accuracy, we show the difference between the real location trace and the estimated location trace.

to clarify two different types of location traces: *real location trace* and *estimated location trace*. A main difference between them is their corresponding timestamps. First, the location trace for user u_n, in general, is a random process $\mathcal{L}^{(n)} = \{\mathcal{L}_t^{(n)} : t \in \mathbb{T}\}$. For the real location trace, the time instants are when there are actually location transitions for u_n, and the corresponding timestamp set is denoted as $\mathbb{T}_r$. The time sequence for the estimated location trace constructed by the adversary is the same set $\mathbb{T}_o$ for the corresponding occupancy trace because the location trace is inferred from the occupancy trace. Usually, $\mathbb{T}_o \neq \mathbb{T}_r$ holds since occupancy detection instants are discrete and periodic, and lagging behind when actual location transitions happen.

In addition, we define $\mathcal{L}_t = (\mathcal{L}_t^{(1)}, \cdots \mathcal{L}_t^{(N)})$ as the collection of the location variables at time t. It takes values from set $\mathbb{L} = \mathbb{L}^{(1)} \times \cdots \mathbb{L}^{(N)}$. The location trace for all users therefore is defined as $\mathcal{L} = \{\mathcal{L}_t : t \in \mathbb{T}\}$.

3.3 Context/Auxiliary Information

Many people claim that occupancy is a privacy-preserving metric mainly because it reveals no identity information. However, occupancy combined with context information discloses private details about indoor locations of building users. A common piece of context information is office directory. This mapping between users and office rooms offers us the opportunity to link an occupancy reading of a room with the presence of a specific user since the one in the office is more likely to be the office owner. The office information can be leveraged to associate estimated location traces to specific users. The role of office directory will be manifest as we describe the intuition of our approach in the next section.

There might also be some past location traces available with which we can establish our prior knowledge about a user's mobility pattern. We will specify in later section how the prior knowledge is established and incorporated.

3.4 User Movement Profiles

A user's movement profile describes the pattern of transitions between different rooms in the building. Essentially, this profile is represented by a Markov transition matrix as shown in Figure 1 with P_{ij} specifying the probability of moving from room r_i to r_j when there is a transition. In our approach, the location trace is modelled as a Markov process.

3.5 Objectives and Evaluation Metrics

The objective specifies what private information the adversary aims to extract from occupancy data. Potential objectives include: *localization* in which the adversary attempts to find out locations of users at certain time instants; *meeting disclosure* in which the adversary is interested in finding out who met whom at a time instant in a given room; or *full tracking* in which the adversary aims to reconstruct the most probable joint location trace for all users [43].

For the localization and meeting disclosure attacks, to evaluate the inference results we can compare the estimated locations with the actual locations of users. In this paper, we focus more on the full tracking attack. To assess the outcome of the attack, we compare the estimated location trace with the real location trace. Noteworthily, we can evaluate the trace for each individual user or the joint trace for all users.

The real location for user u_n at time t is denoted as $l_t^{(n)}$ and the full trace denoted as $l^{(n)}$. We denote as $\hat{l}_t^{(n)}$ the estimated location for user u_n at time t, and the full trace as $\hat{l}^{(n)}$.

As illustrated in Figure 2, by comparing the estimated trace $\hat{l}^{(n)}$ with the real trace $l^{(n)}$ over the time span and finding the percentage of overlap, we can compute the *time-averaged accuracy* $\texttt{ta\_acc}$ of $\hat{l}^{(n)}$:

$$\texttt{ta\_acc}(\hat{l}^{(n)}, l^{(n)}) = \frac{1}{t_T - t_1} \int_{t_1}^{t_T} \mathbb{1}(\hat{l}_\tau^{(n)}, l_\tau^{(n)})\, d\tau,$$

where $\mathbb{1}(x, y)$ is the binary indicator function that equals to 1 when $x = y$ and 0 otherwise.

On the other hand, we may only be interested in the correctness of the estimation of location transitions. In this case, we compare the estimated trace $\hat{l}^{(n)}$ with the sampled location trace. As shown in Figure 2, to obtain the sampled trace, we align location transitions in the real trace to sensing instants. We denote as $l_t^{(n)}[t_s]$ the sampled location for user u_n at time t with sensing interval t_s, and the full sampled trace as $l^{(n)}[t_s]$. Then we can define the *transition-based accuracy* $\texttt{tb\_acc}$ of $\hat{l}^{(n)}$ compared to $l^{(n)}[t_s]$ as:

$$\texttt{tb\_acc}(\hat{l}^{(n)}, l^{(n)}[t_s]) = \frac{1}{T} \sum_{t \in \mathbb{T}_o} \mathbb{1}(\hat{l}_t^{(n)}, l_t^{(n)}[t_s]).$$

Since the location transitions are of interest, we compare the estimated trace with the sampled trace only at instants specified by $\mathbb{T}_o$ as defined in the location trace subsection.

The above definitions of accuracy evaluate the estimated location trace for each individual user. If combining the locations for all users at each time into a N-sized tuple, we can evaluate the joint location trace for all users. The accuracy in this case can be calculated in rather similar way.

4. FROM OCCUPANCY TO LOCATION

In this section, we start with the intuition of applying the factorial hidden Markov model to inferring indoor locations from occupancy. Then we present the model and related technical details in the context of this location inference problem.

4.1 Overview of the Approach

Reconstructing location traces from occupancy data is non-trivial because: 1) occupancy measurements are aggregates of locations, revealing no identities of users in certain rooms, and 2) location traces of multiple users interleave with each other, incurring ambiguity when translating occupancy changes to location transitions of different users. For example, occupancy measurements report that two people are present in room 100, and at next instant there is one person in room 101 and the other in room 102. Each time along with an occupancy change, there is a 'location reshuffle' of users. However, different location transition combinations can lead to the same occupancy change, so occupancy is lossy.

To address the two challenges, we leverage *context information* as well as *user mobility properties*. Specifically in our approach, the context information is the directory that provides office information about users. Even through an occupancy measurement itself is sanitized, directory information can act as a side-channel for inferring underlying identities of occupants. An office can thus serve as a user's identifier. The one sitting in an office is most likely to be the office owner. The context information can be captured by the mobility model as we will explain later.

As previously mentioned, an occupancy change may lead to multiple location transition combinations. The adversary needs to find the correct combination. Without explicitly tracking users, deterministic approaches prove infeasible. Fortunately, the location reshuffle is not uniformly random over all combinations, and users' mobility patterns can be leveraged to determine the preferences over them. Hence, the mobility pattern enables attributing tangled transitions to specific users. Each user's location transitions can be modelled as a first-order Markov process, which proves feasible in describing user mobility patterns [46, 47]. In addition, this first-order assumption makes our analysis easier and the inference algorithm simpler while still yielding satisfactory inference results as we will demonstrate in Section 7. Another concern regarding the Markov transition model may arise if a user exhibits different patterns during different time periods within a day. This drawback can be mitigated by introducing a heterogeneous Markov model to capture one's movement patterns during different periods in a day.

The context information can be incorporated into the Markov model since a user's location transitions are usually centred on his office. As we can observe from a real dataset, over 50% of transitions are either from or to the office. Hence, the Markov transition matrices for different users are guaranteed to be distinct if users have different offices. The discrepancy enables differentiating various users and attributing occupancy changes to location transitions. If multiple users share an office, their transition matrices might or might not be different enough to distinguish any of them. The feasibility hinges on the discrepancy between transition matrices, which is not necessarily present in all real scenarios. The degree of discrepancy is affected by working environment, personal habits and many other factors. Hence, a definitive conclusion would require a comprehensive study of human mobility behavior in various indoor environments, which is beyond the scope of this paper. Instead, we focus on the problem of modelling the relationship between the users' mobility patterns and the occupancy data to demonstrate the ability to infer user activities. Given user locations, we can obtain occupancy through deterministic aggregation. The reverse process, however, is probabilistic and complicated, and its success relies on the information loss in the location-to-occupancy process and the inherent entropy in user location transitions over time

To give an example, if we observed two users in one room and then one of them left the room and entered another room, we are unable to find which one of the two made this transition by this occupancy change. However, if the one who left entered an office, the user can be identified with high probability based on the ownership of the office. This explains how context information helps in removing ambiguity. The context information can be incorporated into the Markov transition matrix such that a user has higher probability of returning his office from other rooms than other users going to his office.

Returning to the discussion about the intuition behind our approach, the occupancy traces are generated by the occupancy detection system. Occupancy at a time only relies on user locations at that instant. The characteristics of the detection system determines the relation between user locations and room occupancy. The relation can be deterministic when detection is accurate or probabilistic if there is random noise. The detection system characteristics can be investigated beforehand.

Taking into account the Markovian property of location traces and the relation between occupancy and location, we can hence describe the inference framework using the hidden Markov model (HMM) [48]. In a HMM, the current state of the process cannot be observed directly. Instead, some outputs from the current state can be observed. The probability distribution of the outputs depends only on the current state. Specifically in our problem, we apply the factorial hidden Markov model (FHMM) instead of the HMM. The FHMM offers us several benefits as we will discuss soon.

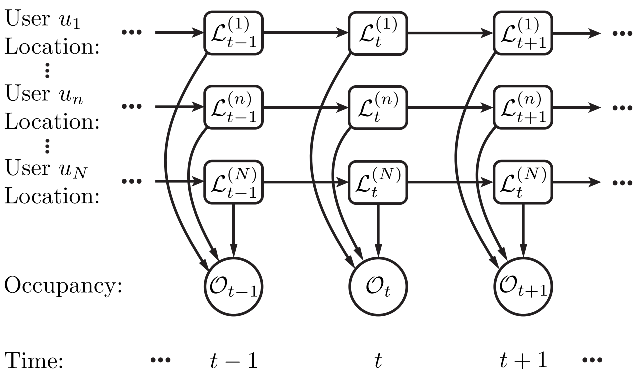

Figure 3: We illustrate the FHMM model used for location inference. The model consists of multiple latent state chains each of which represents the location trace of one user. The observation sequence shows occupancy changes for all locations.

4.2 Factorial Hidden Markov Model

In the HMM, locations of all users are combined into one latent variable. All user locations are allowed to interact arbitrarily. The advantage is the model can incorporate possible interactions of location transitions among users. However, the drawbacks render HMM infeasible. The size of the state space for the single latent variable is K^N, and leads to intolerable computational complexity. Additionally, it requires more learning data for the purpose of establishing the Markov transition matrix. Hence we resort to FHMM which decouples location transitions of different users. The location traces for different users are assumed to be independent from each other. They are linked together by the occupancy trace. The state space is thereafter reduced to KN. The computational efficiency can be significantly improved. Even though the independence assumption might not hold in some scenarios, it makes the algorithm trackable without undermining the inference accuracy.

As illustrated in Figure 3, the sequence of observations is the *occupancy trace*. The FHMM consists of multiple latent state sequences, each of which represents the *location trace* of a user. Each location trace follows its own Markovian transition nature as users exhibit different mobility patterns. At each time instant, the occupancy measurement depends on locations of all users so we can see arrows to the observation from corresponding latent states in Figure 3. Two fundamental problems for FHMM are *learning* and *inference*. We will explain them later in the context of our scenario.

4.3 Model Parameters

The model is governed by a set of parameters: initial state probabilities, transition probabilities and emission probabilities. In the context of our scenario, initial state probabilities specify the chance of each user starting from certain locations. Since every user is initially located outside the building, with probability 1 the initial latent state is `away`. Initial state probabilities are invariant.

The transition probabilities specify the mobility pattern of a user, which is denoted as a $K \times K$ transition matrix. We define as $\mathbf{A}^{(n)} = [a_{ij}^{(n)}]$ $(i, j = 1, 2, \cdots, K)$ the transition matrix for user

u_n, where $a_{ij}^{(n)} = P(\mathcal{L}_{t+1}^{(n)} = r_j | \mathcal{L}_t^{(n)} = r_i)$ for $t = 1, 2, \cdots, T-1$ as the transition model is homogeneous such that transition probabilities do not change over time.

The emission probabilities characterize the conditional distribution of occupancy measurements given user locations, defined as $P(\mathcal{O}_t|\mathcal{L}_t)$. The conditional distribution hinges on specific occupancy detection systems. In this paper, we consider the worst case of privacy leakage in which occupancy is accurately detected. Given precise occupancy information, location inference can achieve better performance. This gives us an upper bound on the potential privacy risk. The conditional distribution hence becomes degenerate and gives the correct occupancy.

The learning problem in HMM is to establish model parameters. In our model, we need to learn the transition matrix for all users. We can denote the model parameters as $\lambda = (\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)})$. Input to the learning module includes the occupancy trace, context information and possibly some past location trace samples.

5. KNOWLEDGE CONSTRUCTION

In order to infer locations from occupancy, we first need to establish the knowledge about user mobility patterns. Knowledges construction, in the context of the FHMM, translates into learning the model parameters, namely the transition matrices for all users. We present two methods for parameter estimation.

5.1 Learning Transition Matrices

The model parameter can be established using the maximum likelihood estimation (MLE). Formally, we need to find the parameter $\lambda^\star$ that maximizes the probability of the observation (occupancy) sequence, i.e.: $\lambda^\star = \text{argmax}_\lambda\ P(\mathcal{O}|\lambda)$. Solving the optimization problem involves the expectation maximization (EM) algorithm, which is known as the Baum-Welch algorithm in the HMM [48]. The EM algorithm iterates between two steps commonly referred to as E and M. In the E step, we compute the posterior distribution over latent states using current parameters, and obtain the expected log-likelihood of observations as a function of parameters. In the M step, we maximize the expectation and update the parameters. The EM algorithm for MLE is defined as:

$$Q\left(\lambda^{(i+1)}|\lambda^{(i)}\right) = E_{\mathcal{L}|\mathcal{O},\lambda^{(i)}}\left[\log P(\mathcal{L}, \mathcal{O}|\lambda^{(i+1)})\right],$$

where $\lambda^{(i)}, \lambda^{(i+1)}$ are the parameters at the i^{th} and $(i+1)^{th}$ iteration. The Q function takes conditional expectation over all possible location traces and is maximized with respect to $\lambda^{(i+1)}$. The E step requires computing posterior probabilities $P(\mathcal{L}|\mathcal{O}, \lambda^{(i)})$, which incurs intensive computation for exact inference. Hence, Gibbs sampling is applied to generate location traces and approximate the posterior distributions [23].

Each iteration of EM samples all TN latent location variables. After adequate iterations, the samples can approximate the posterior probabilities. Then in the M step, we can update the transition matrix using the obtained posterior probabilities

$$a_{ij}^{(n)} = \frac{\sum_{t=1}^{T-1} P(\mathcal{L}_t^{(n)} = r_i, \mathcal{L}_{t+1}^{(n)} = r_j|\mathcal{O}, \lambda)}{\sum_{t=1}^{T-1} P(\mathcal{L}_t^{(n)} = r_i|\mathcal{O}, \lambda)}.$$

The algorithm iterates forth and back between the two step until convergence of the log-likelihood function. We thereafter obtain the estimation of the user movement profiles.

5.2 Prior Knowledge

The approach described above leverages no prior knowledge that we may gain from auxiliary information. Past location traces, if available, provides us the prior knowledge about users' mobility patterns. For the transition matrix $\mathbf{A}^{(n)} = [a_{ij}^{(n)}]$ $(i, j = 1, 2, \cdots, K)$, we assume that the rows of $\mathbf{A}$ are independent and their densities follow Dirichlet distributions. By tuning the parameters of a certain Dirichlet distribution, we are able to generate various density functions for a row vector in $\mathbf{A}^{(n)}$. A transition matrix is therefore governed by K Dirichlet distributions. In such a way, we can create various movement profiles for each user. Considering the independence between rows of a transition matrix and between the matrices of different users, the density of the model parameters becomes

$$P(\lambda) = C \cdot \prod_{n=1}^{N} \left(\prod_{j=1}^{K} \left(\prod_{i=1}^{K} \left(a_{ij}^{(n)} \right)^{\eta_{ij}^{(n)}-1} \right) \right),$$

where C is the normalizing factor, $[\eta_{ij}^{(n)}]$ $(i = 1, \cdots, K)$ is the set of parameters of the Dirichlet distribution for the j^{th} $(j = 1, \cdots, K)$ row of the transition matrix $\mathbf{A}^{(n)}$.

Given past location traces of a user n, we can obtain the set of parameters for the Dirichlet distributions. The parameter $\eta_{ij}^{(n)}$ equals to the number of location transitions of user n from r_i to r_j. The parameters represent the prior knowledge of movement patterns, and is taken into account when estimating the location transition probabilities. The method is called maximum a posterior (MAP) estimation, defined as $\lambda^\star = \text{argmax}_\lambda\ P(\lambda|\mathcal{O}) = \text{argmax}_\lambda\ P(\mathcal{O}|\lambda)P(\lambda)$. Similarly, we need to apply the EM algorithm for MAP which introduces the additional term $P(\lambda^{(i+1)}$ compared to the version for MLE, yielding

$$Q'\left(\lambda^{(i+1)}|\lambda^{(i)}\right) = Q(\lambda^{(i+1)}|\lambda^{(i)}) + \log P(\lambda^{(i+1)}),$$

where we can observe an additional term of the logarithm density of model parameters compared to the Q function for MLE. The posterior probabilities are also approximated using Gibbs sampling. The resulting update for the transition matrix in the M step becomes

$$a_{ij}^{(n)} = \frac{\sum_{t=1}^{T-1} P(\mathcal{L}_t^{(n)} = r_i, \mathcal{L}_{t+1}^{(n)} = r_j|\mathcal{O}, \lambda) + \eta_{ij}^{(n)} - 1}{\sum_{t=1}^{T-1} P(\mathcal{L}_t^{(n)} = r_i|\mathcal{O}, \lambda) + \sum_{j=1}^{K}(\eta_{ij}^{(n)} - 1)}.$$

Each update incorporates the transitions seen in past location traces and hence forces the estimation to lean to prior knowledge in contrast to the MLE estimation.

6. LOCATION INFERENCE ATTACKS

Leveraging the transition matrices of all users we obtained, we are able to mount several types of location inference attacks. Recall from the objectives, the adversary aim to infer locations at certain time instants or reconstruct the whole trace.

6.1 Localization and Meeting Disclosure Attacks

In localization attacks, the attacker aims to find the location of a user at a specific time instant. Formally, we need to compute $P(\mathcal{L}_t^{(n)} = r_i|\mathcal{O}, \lambda)$, which specifies the distribution of the location of user n at time t. The probability can be computed using the modified forward-backward algorithm as proposed by Ghahramani et al. [23]. The forward algorithm calculates the probability of locations for all users at a certain time and the occupancy sequence up to that time, defined as $\alpha_t = P(\mathcal{L}_t^{(1)}, \cdots, \mathcal{L}_t^{(N)}, \mathcal{O}_1, \cdots, \mathcal{O}_t|\lambda)$. The backward algorithm computes the probability of future occupancy trace conditioned on current locations of all users, defined as $\beta_t = P(\mathcal{O}_{t+1}, \cdots, \mathcal{O}_T|\mathcal{L}_t^{(1)}, \cdots, \mathcal{L}_t^{(N)}, \lambda)$.

Given the forward and backward variables at time t, we can compute the posterior probabilities at that time. Summing over locations of other users, we can obtain the marginal probability of the location for user n:

$$P(\mathcal{L}_t^{(n)}|\mathcal{O},\lambda) = \sum_{\mathcal{L}_t^{(m)}:m\neq n} P(\mathcal{L}_t^{(1)},\cdots,\mathcal{L}_t^{(N)}|\mathcal{O},\lambda).$$

In a meeting disclosure attack, the attacker aims to determine whether a pair of users u_n and u_m met in room r_i at time t. Formally, the attacker needs to compute $P(\mathcal{L}_t^{(n)} = r_i, \mathcal{L}_t^{(m)} = r_i|\mathcal{O},\lambda)$, which can be computed by marginalizing all other variables:

$$P(\mathcal{L}_t^{(n)},\mathcal{L}_t^{(m)}|\mathcal{O},\lambda) = \sum_{\mathcal{L}_t^{(l)}:l\neq n,m} P(\mathcal{L}_t^{(1)},\cdots,\mathcal{L}_t^{(N)}|\mathcal{O},\lambda).$$

Depending on different objectives, we can extend the meeting disclosure attacks to consider more than two users at multiple instants.

6.2 Location Tracking Attack

In location tracking attack, the objective is to reconstruct the most probable location trace given the occupancy trace. Formally, we need to find $\hat{l} = \text{argmax}_l\ P(\mathcal{L} = l|\mathcal{O},\lambda)$. The approach to this problem is well known as the Viterbi algorithm in HMM. In our context, a straightforward approach is to treat the locations of all users as a single latent variable and directly apply the Viterbi algorithm. The approach, however, is infeasible as the state space for the location variable is K^N.

We modify the Viterbi algorithm to reconstruct the most likely location trace in FHMM. Considering that the location traces are independent, we can sequentially advance the location trace for each user to next time instant. The state space is reduced to KN and the computational overhead is significantly reduced.

For the ease of presentation, we denote as $\{\mathcal{L}.\}_p^q$ the location sequence $\mathcal{L}_p,\cdots,\mathcal{L}_q$. Similarly, $\{\mathcal{O}.\}_p^q = \mathcal{O}_p,\cdots,\mathcal{O}_q$ and $\{\mathcal{L}_t^{(\cdot)}\}_m^n$ $= \mathcal{L}_t^{(m)},\cdots,\mathcal{L}_t^{(n)}$. We can then define the recursion variables

$$\begin{aligned}
\phi_t &= \max_{\{\mathcal{L}.\}_1^{t-1}} P(\{\mathcal{L}.\}_1^{t-1},\{\mathcal{L}_t^{(\cdot)}\}_1^N,\{\mathcal{O}.\}_1^t) \\
\phi_t^{(0)} &= \max_{\{\mathcal{L}.\}_1^{t-1}} P(\{\mathcal{L}.\}_1^{t-1},\{\mathcal{L}_t^{(\cdot)}\}_1^N,\{\mathcal{O}.\}_1^{t-1}) \\
\phi_t^{(1)} &= \max_{\{\mathcal{L}.\}_1^{t-2},\{\mathcal{L}_{t-1}^{(\cdot)}\}_2^N} P(\{\mathcal{L}.\}_1^{t-2},\{\mathcal{L}_{t-1}^{(\cdot)}\}_2^N, \\
&\qquad \mathcal{L}_{t-1}^{(1)},\{\mathcal{L}_t^{(\cdot)}\}_2^N,\{\mathcal{O}.\}_1^{t-1}) \\
&\vdots \\
\phi_t^{(N)} &= \max_{\{\mathcal{L}.\}_1^{t-2}} P(\{\mathcal{L}.\}_1^{t-2},\{\mathcal{L}_{t-1}^{(\cdot)}\}_1^N,\{\mathcal{O}.\}_1^{t-1}).
\end{aligned}$$

Here, ϕ_t is a function of $\mathcal{L}_t$, which gives the maximum probability of location traces ending with $\mathcal{L}_t$. Different from the original Viterbi algorithm, the modified algorithm has a sequence of intermediate variables which update the location of each user to the next time instant. Intermediate variable $\phi_t^{(n)}$ is a function of a combination of locations at time t and locations at time $t-1$, or precisely $\left\{\{\mathcal{L}_{t-1}^{(\cdot)}\}_1^n,\{\mathcal{L}_t^{(\cdot)}\}_{n+1}^N\right\}$. We can therefore obtain recursion relations as below

$$\phi_t = \phi_t^{(0)}P(\mathcal{O}_t|\mathcal{L}_t) \tag{1}$$

$$\phi_t^{(n-1)} = \max_{\mathcal{L}_{t-1}^{(n)}} \phi_t^{(n)}P(\mathcal{L}_t^{(n)}|\mathcal{L}_t^{(n)}) \tag{2}$$

$$\phi_t^{(N)} = \phi_{t-1}(\mathcal{L}_{t-1}). \tag{3}$$

For this factorial model, we update the location for each user independently and sequentially as specified by (2).

To construct the most probable location trace, two kinds of backtracking procedures are performed: local and global. The local backtracking finds the most probable joint locations for all users at a time instant, namely $\mathcal{L}_t$. The global backtracking finds the most probable location trace over time, namely $\{\mathcal{L}.\}_1^T$. The update from ϕ_{t-1} to ϕ_t includes N steps specified by (2). Each step we store the previous location of user n that leads to the maximum probability

$$\Phi_t^{(n-1)}(\mathcal{L}_t^{(n)}) = \underset{\mathcal{L}_{t-1}^{(n)}}{\text{argmax}}\ \phi_t^{(n)}P(\mathcal{L}_t^{(n)}|\mathcal{L}_{t-1}^{(n)}).$$

Then by local backtracking, we can progressively obtain the optimal location for each individual user and eventually construct the most probable joint locations for all users at time t. The procedure is defined as

$$\mathcal{L}_{t-1}^{\star(1)} = \underset{\mathcal{L}_{t-1}^{(1)}}{\text{argmax}}\ \phi_t^{(1)}, \quad \mathcal{L}_{t-1}^{\star(n)} = \Phi_{t-1}^{(n-1)}(\mathcal{L}_{t-1}^{\star(n-1)})$$

for $n = 1,\cdots,N$. At each time t, we store the previous joint locations of all users that result in the maximum probability

$$\Phi_t(\mathcal{L}_t) = \text{argmax}_{\mathcal{L}_{t-1}}\ \phi(\mathcal{L}_t)P(\mathcal{L}_t|\mathcal{L}_{t-1}).$$

Then applying global backtracking which is similar to the backtracking in Viterbi algorithm, we start from the last time instant and reconstruct the location trace backward.

7. EXPERIMENTAL STUDY

In this section, we evaluate our inference approach. The metric used to quantify the inference performance is *accuracy*. The purpose of assessing the inference method is to demonstrate the possibility of inferring locations from occupancy data in certain environments and to reveal the underlying privacy risk. Moreover, we hope our results can provide insights into design and implementation of privacy-preserving sensing systems. Toward our goal of privacy-preserving sensing system design, we study two impacting factors: the sensing interval and the number of users.

7.1 Dataset

To setup our experiment, we use synthetic data as well as real-world data from the the *Augsburg Indoor Location Tracking Benchmark* [45]. The dataset includes location traces for 4 users in an office building with 14 rooms. The benchmark dataset contains location data over a period of 2 to 9 weeks. Each trace records location transitions of one user during a day. All traces are timestamped. The Augsburg dataset is the only suitable indoor room-level location dataset we found publicly available. Though relatively small in scale, the location traces in the dataset are representative of the transition patterns of many people in certain working environments.

As previously discussed, in many scenarios the office room assumes an important role in a user's transitions. In Table 1, we show two statistics about the Augsburg dataset. Noteworthily, of all transitions per day, 52% to 70% are either originate from or destined to one's office, and office can represent one's identity. Each user's transitions are heavily linked to his office, resulting in discrepancy in their transition matrices.

To investigate the impact of the population, we create synthetic data that simulates location traces for 20 users based on the Augsburg dataset. To simulate the location traces, we apply the random way-point mobility model. A user randomly chooses the next location according to his movement profile, i.e. the Markov transition matrix. The stay time at one location is normally distributed.

Table 1: We show the average number of transitions each user made per day, and the average percentage of transitions from or to one's office.

User	avg num of trans per day	avg percent of trans from/to office per day
1	26.2	52.5%
2	37.7	70.7%
3	35.2	66.8%
4	26.6	65.7%

To precisely simulate the transitions, we also take into account the walking speed and the distance between rooms.

All the parameters for the mobility model are created based on the empirical study of the Augsburg dataset. Tweaking the existing movement profiles, we create transition matrices for another 16 users. The stay time in a room varies for different users and rooms. Typically, the distribution for one's office has larger mean values and larger variance, while the distribution for other rooms has smaller mean and variance.

7.2 Methodology

First, we experiment with the benchmark data. We study the impact of the sensing interval. We aim to infer location traces of the 4 users under different sensing intervals. Though the dataset provides location traces over a period of 2 to 9 weeks, there were 10 days when all 4 users were present in the building. The location traces in the 10 days exhibit differences in the number of transitions, the total length and the stay time at rooms. We assume the detection system can accurately detect occupancy at each room. Hence, given the location traces of all users, we can obtain the occupancy traces for all rooms by a simple counting. We apply different intervals and get occupancy traces of different granularities. For each sensing interval, the resulting occupancy trace is used to perform 10 fold cross-validation. Specifically, to perform the cross-validation we partition the location traces and the corresponding occupancy trace into 10 complementary subsets. One round of validation involves performing training on 9 subsets (called the training set), and validating the model on the remaining subset (called the testing set). To reduce variability, cross-validation is conducted using different training sets, and the validation results are averaged over 10 rounds.

Second, we study the impact of population using the synthetic data. The sensing interval is fixed to 1 second to show the optimal inference performance. We create scenarios of different numbers of users from 1 to 20. Each time, we can obtain the occupancy traces from all location traces. Similarly as the experiment with the Augsburg data, we perform 10 fold cross-validation over the synthetic data of 10 days. As the interval is 1 second, the `tb_acc` and the `ta_acc` becomes equivalent. We compare the estimated location traces with real location traces and evaluate the accuracy.

We conduct our experiment based on available sample traces. To also show the statistical significance, we evaluate our estimation of the mean value of accuracy. We assume that the accuracy in each iteration of the cross-validation is a normally distributed random variable. Denote as X_i $(i = 1, \cdots, 10)$ the accuracy result obtained in the i^{th} run of the cross-validation, and $X_i \sim N(\mu, \sigma^2)$. The sample mean $\overline{X}$ follows normal distribution. The sample variance S^2 follows chi-square distribution. Then we can obtain that $U = \frac{\overline{X}-\mu}{S/\sqrt{10}} \sim t(9)$, where the random variable U follows t-distribution with the degree of freedom 9. Therefore, we are able to obtain that $P(|\overline{X} - \mu| \leq 0.03) = P(|U| \leq \frac{0.03}{S/\sqrt{10}})$, which essentially computes the probability that the estimated mean is within a small bound around the true mean value. This probability describes the confidence in estimations of the accuracy on average.

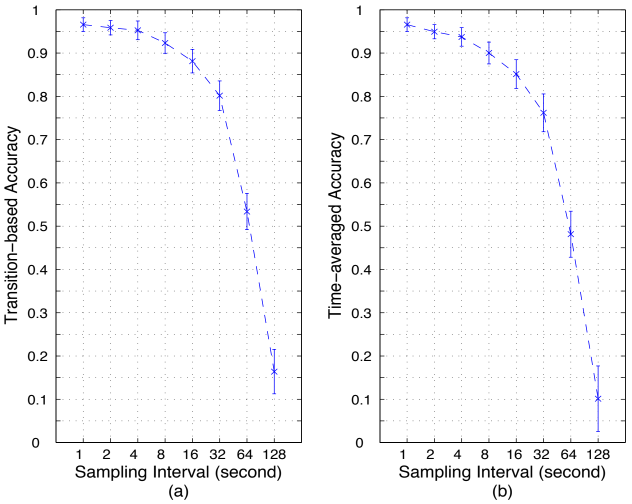

Figure 4: We show the results of the 10 fold cross-validation. The number of users is 4. The errorbar plot in (a) and (b) illustrates the mean and standard deviation of transition-based accuracy and time-averaged accuracy, respectively, under sensing intervals from 1 to 128 seconds.

7.3 Accuracy and Sensing Interval

Figure 4 illustrates the results about the impact of the sensing interval, and the difference between transition-based accuracy and time-averaged accuracy. Figure 4(a) shows the performance using metric `tb_acc` while Figure 4(b) shows the result for `ta_acc`. In both figures, the sensing interval starts from 1 second and increases to 128 seconds. For each sensing interval, the errorbar plot depicts mean and standard deviation of the 10 fold cross-validation.

In Figure 4(a), when the sensing interval is 1 second, `tb_acc` can achieve 0.96 on average. As the sensing interval increases to 128 seconds, the mean gradually decreases to below 0.2. The standard deviation slightly increases.

In Figure 4(b), the errorbar plot for `ta_acc` exhibits similar trend as the plot for `tb_acc`. When the sensing interval is 1 second, `ta_acc` becomes `tb_acc`. It reaches about 0.96. As the sensing interval increases to 128 seconds, `ta_acc` slides to about 0.1 on average. With a larger sensing interval, the occupancy detection system might miss certain occupancy changes and render the inference system fail to reconstruct certain location transitions that led to those changes. In addition, a larger sensing interval results in imprecision in timestamps. Comparing figure (b) with figure (a), we can find that `tb_acc` outperforms `ta_acc`by a margin that expands as the sensing interval increases.

Figure 6(a) shows the t-analysis result for the transition-based accuracy. We evaluate the probability that the sample mean of the transition-based accuracy is within ±0.03 of the real mean value. The probability maintains 1 when the sensing interval is below 16 seconds, and declines to about 0.98 as the sensing interval increases to 128 seconds. This result agrees with the Figure 6(a) as the variance increases with the sensing interval. The estimation therefore becomes less accurate. Figure 6(b) shows the t-analysis result for the real accuracy. The curve exhibits similar trend. It is, by con-

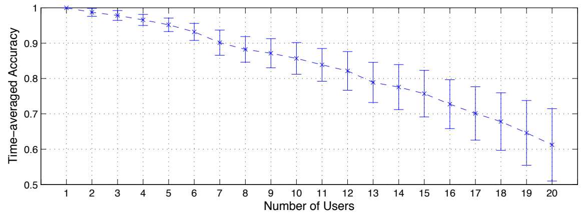

Figure 5: We show the 10 fold cross-validation results in the presence of different numbers of users from 1 to 20. The errorbar plot illustrates the mean and standard deviation of inference accuracy. The sensing interval is fixed at 1 second. Transition-based accuracy and time-averaged accuracy become interchangeable in this case.

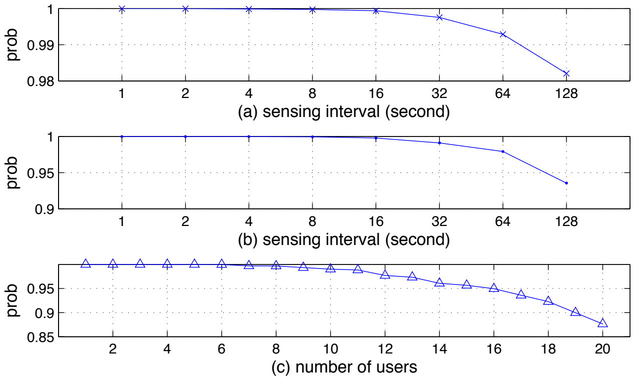

Figure 6: We show through *t*-analysis the probability that the sample mean of accuracy is within ±0.03 of the actual mean value. Figure (a) shows the result about the transition-based accuracy in Figure 4(a). Figure (b) shows the result about the time-averaged accuracy shown in Figure 4(b). Figure (c) shows the result about the accuracy illustrated in Figure 5.

trast, smooth as the variance in Figure 6(b) does not increase significantly with the sensing interval.

7.4 Accuracy and Number of Users

Figure 5 illustrates the results that we have obtained about the impact of the number of users. The sensing interval is chosen to be 1 second. The number of users increases from 1 to 20. In each case, the errorbar plot depicts the mean value and the standard deviation of the inference accuracy. Since the transition-based accuracy and the time-averaged accuracy becomes interchangeable when the sensing interval is 1 second, we do not distinguish between these two metrics. When there is only one user, occupancy is equivalent to location, which leads to fully correct estimation of location trace of that user. As the number of users increases, the location traces of different users interleave with each other, which leads to uncertainty in reconstructing the location traces and inaccuracy in estimation. The inference accuracy on average declines to about 0.61 when there are 20 users. The variance is gradually increasing with the number of users, which also supports the fact that more users incur more uncertainty. Figure 6(c) shows the t-analysis result. The confidence of estimation is close to 1 with less than 8 users. It slides in case of more users since the variance climbs.

7.5 Average Accuracy and Joint Accuracy

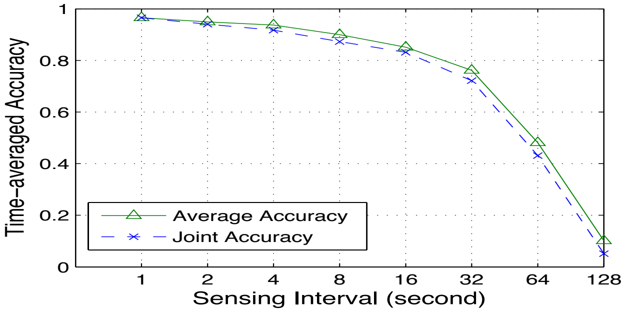

Figure 7: The two curves depict the mean value of joint accuracy and average accuracy, respectively, as the sensing interval increases from 1 to 128 seconds.

As explained when defining the metrics, the metrics used average the accuracy of all location traces. We call it *average accuracy*. Figure 7 compares the average accuracy with *joint accuracy* that denotes the accuracy of the joint location trace for all users. In the joint location trace, locations of all users at a time are recognized as a tuple, and hence a mistake for even one user renders the whole tuple incorrect. The joint accuracy provides a lower bound on the accuracy for individual users. From the figure, we can observe a non-negligible difference between the joint accuracy and the average accuracy. When the sensing rate is 1 second, the average accuracy outperforms by a small margin of about 0.01. The difference expands to 0.05 as the sensing interval increases.

8. SENSING SYSTEM DESIGN

The sensing system can leverage the results of our empirical study to achieve adaptive control of privacy. The accuracy metric we use measures the loss of location privacy. Accuracy is affected by several factors, among which the sensing interval is a configurable system parameter while the number of users is a context factor that the system cannot change. We discuss how the system takes into account multiple factors and performs adaptive control.

8.1 Design Metrics

The sensing system feeds sensor measurements to a variety of service providers residing above it to enable automatic building management. A central goal of the sensing system is thus to provide adequate data to data consumers so that they can perform their functionalities properly. In other words, the sensing system needs to provide sufficient *data utility*.

There are several system parameters affecting data utility: sensor deployment density, sensing interval, and measurement accuracy. The parameters can be configured to achieve different operation goals. Other than the configuration parameters, the utility is also impacted by the context including the population in the building, the behavior patterns of the users and the topology of the building. We incorporate impacting factors and the context information with a single utility function $\texttt{Util}_{\mathcal{C}}(\mathcal{P})$ in which $\mathcal{P}$ denotes relevant parameters that the system can configure, and $\mathcal{C}$ represents the contextual settings that the system cannot really change yet might be aware of through sensing, such as the population in the building.

The deployment density represents the spatial granularity of the sensing system. In general, with more sensors the system provides more utility because more area is under coverage. The sensing interval represents the temporal granularity of the system. More frequent sensing enables service providers to have a finer temporal granularity in their view of the building environment, and allows the building management system to respond to environmental variations properly and promptly. The measurement accuracy represents the inherent quality of sensor readings, i.e. how close the measurement is to the real physical properties such as temperature, humidity, illumination and so forth. The utility function $\mathtt{Util}_{\mathcal{C}}(\mathcal{P})$ is an increasing function of the parameters described above.

On the other hand, system parameters also affect *user privacy*. In order to evaluate the sensing system with respect to protecting private details of building occupants, we need a privacy function $\mathtt{Priv}_{\mathcal{C}}(\mathcal{P})$ which characterizes to what extent the system preserves the specific privacy of interest. The context information $\mathcal{C}$ also affects our assessment of privacy. For example, when there are more users in the building, the resulting higher entropy leads to more privacy for the users. Considering the system parameters mentioned above, higher deployment density leads to less privacy for users. Smaller sensing interval also allows attackers to infer user activities with higher accuracy and confidence. Likewise, increasing measure accuracy offers more opportunities to perform effective inference. In addition to these parameters, in our indoor location inference scenario $\mathtt{Priv}_{\mathcal{C}}(\mathcal{P})$ also depends on the number of users and their mobility patterns. The number of users represents inherent complexity of the tracking problem. In the case of only one user, location inference becomes trivial as occupancy amounts to the location of that user. However, when more users are present, it becomes generally more difficult to reconstruct location from occupancy since location traces of different users interleave with each other. The privacy function is a decreasing function of the parameters, which essentially presents a tradeoff with the utility function.

8.2 Adaptive Control

The goal of the sensing system is to achieve privacy-aware and context-aware adaptive control. In what follows, we discuss how the system can be configured under various contextual settings to operate in accordance to the utility-privacy tradeoff and specific requirements for performance.

Formally, given the utility function $\mathtt{Util}_{\mathcal{C}}(\mathcal{P})$ and the privacy function $\mathtt{Priv}_{\mathcal{C}}(\mathcal{P})$, we can formulate an optimization problem by combining two functions into a single objective function according to adjustable design goal

$$\mathcal{P}^{\star} = \mathrm{argmax}_{\mathcal{P}} \left(u\mathtt{Util}_{\mathcal{C}}(\mathcal{P}) + v\mathtt{Priv}_{\mathcal{C}}(\mathcal{P}) \right), \qquad (4)$$

where $u, v \geq 0$ are weighting constants specifying the system's emphasis over utility and privacy.

We study the indoor location privacy with respect to sensing interval and number of users. In the presence of different population in the building, by controlling the sensing interval, the system can achieve different operation goals with respect to utility and privacy.

8.3 An Example

As a proof-of-concept example, we discuss how the occupancy detection system can operate under the adaptive control framework. In the last section, we present our results about inference accuracy. As explained, inference accuracy and user location privacy are two sides of the same coin. We can provide an empirical location privacy function computed using our results $\mathtt{Priv}_{\mathcal{C}}(\mathcal{P}) = 1 - \mathtt{tb\_acc}$. The parameter $\mathcal{P}$ is the sensing interval in this example, and the context $\mathcal{C}$ is the number of users.

Due to the absence of a formal study on utility under different sensing intervals and population, we construct the utility function according to several intuitions. For the purpose of illustrating the utility-privacy tradeoff and adaptive control, we simplify some details, and focus on high-level ideas. Though this carefree attitude might be controversial, we hope it can serve as the first dialogue on this topic. In spite of various interpretations of utility, we concentrate on energy efficiency. The intuitions about the utility function include but are not limited to the following.

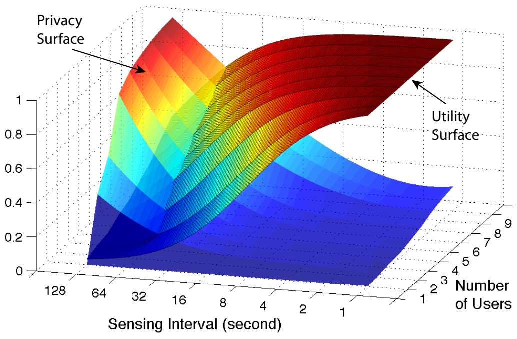

Figure 8: We depict the surface of privacy which has higher values with larger sensing intervals, and the surface of utility which has value 1 when the sensing interval is 1. The surfaces also evolve with the other dimension: the number of users. The system can adjust the sensing interval to achieve desired tradeoff between utility and privacy.

1. Utility is normalized to 0 to 1.
2. Smaller sensing interval leads to more utility. There is a limiting property about the utility the system can achieve.
3. The utility versus sensing interval relation changes with different numbers of users. With more users, the system inherently enjoys more information and more data utility.

We adopt the generalized logistic function to characterize data utility. In Figure 8, we plot the privacy function and utility function with respect to different contexts (number of users) and system parameters (sensing interval). Given an optimization goal defined in (4), the system can can continually adjust its sensing interval to achieve the desired tradeoff as the number of users changes.

9. CONCLUSIONS AND FUTURE WORK

We have demonstrated that time-series occupancy data collected for environmental control in smart buildings leaks privacy-sensitive information about the building users. Toward quantitative privacy assessment, we propose a stochastic framework using the factorial hidden Markov model that captures the characteristics of user mobility and occupancy measurement. In our experiment with the real and synthetic data, we can achieve high inference accuracy with fine-grained occupancy data. More importantly, we show quantitatively the decrease of inference accuracy with respect to two factors in the sensing system: sensing interval and number of users. Based on our framework, the system can pursue fine privacy control by adjusting the sensing interval with respect to the number of users.

While our approach demonstrates the privacy leakage in smart buildings through data analytics, our results only scratch the surface of the potential privacy issues in heavily-sensed environments. The wealth of data collected in such environments goes far beyond occupancy data, suggesting that additional privacy risks exist and must be incorporated into the design of privacy-respecting management systems. Another valuable direction of further study is to conduct extensive experimentation and data collection at scale to validate our belief that the proposed techniques can scale to fit practical scenarios.

10. REFERENCES

[1] A. Rowe, M.E. Berges, G. Bhatia, E. Goldman, R. Rajkumar, J.H. Garrett, J.M.F. Moura, L. Soibelman, "*Sensor Andrew: Large-scale campus-wide sensing and actuation*," IBM Journal of Research and Development, vol.55, no.1.2, pp.6:1,6:14, 2011.

[2] N. Xu, "*A survey of sensor network applications*," IEEE Communications Magazine, 2002.

[3] D. Bourgeois, C. Reinhart and I. Macdonald, "*Adding advanced behavioural models in whole building energy simulation: a study on the total energy impact of manual and automated lighting control*," Energy and Buildings, 38(7):814-823, July 2006.

[4] T. Nguyen and M. Aiello, "*Energy intelligent buildings based on user activity: A survey*," Energy and Buildings, no. 56, pp. 244-257, January 2013.

[5] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin and J. Stankovic, "*ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring*," Technical Report CS-2006-11, Department of Computer Science, University of Virginia, 2006.

[6] **`http://enlightedinc.com/solutions/products/`**

[7] **`http://redwoodsys.com/solutions`**

[8] K. Framling, I. Oliver, J. Honkola, J. Nyman, "*Smart spaces for ubiquitously smart buildings*," the IEEE Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'09), 2009.

[9] J. Kleissl, Y. Agarwal, "*Cyber-physical energy systems: focus on smart buildings*," the 47th Design Automation Conference (DAC'10), New York, NY, USA, 749-754, 2010.

[10] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, T. Weng, "*Occupancy-driven energy management for smart building automation*," BuildSys'10, pages 1-6, New York, NY, USA, 2010.

[11] S.K. Ghai, L.V. Thanayankizil, D.P. Seetharam, D. Chakraborty, "*Occupancy detection in commercial buildings using opportunistic context sources*," the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM'12), 2012.

[12] V.L. Erickson, A.E. Cerpa, "*Occupancy based demand response HVAC control strategy*," BuildSys'10, pages 7-12, New York, NY, USA, 2010.

[13] J. Scott, A. J. B. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar, "*Preheat: controlling home heating using occupancy prediction*," the 13th international conference on Ubiquitous computing (UbiComp'11), New York, NY, USA, 2011.

[14] V.L. Erickson, S. Achleitner, A.E. Cerpa, "*POEM: power-efficient occupancy-based energy management system*," the 12th international conference on Information processing in sensor networks (IPSN'13), New York, NY, USA, 2013.

[15] F. Manzoor, Z. Cong, P. Stack and K. Menzel, "*Tracking occupants and inventory items in buildings using RFID technology*," the 18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering, Weimar, Germany, July 2009.

[16] R. Melfi, B. Rosenblum, B. Nordman, K. Christensen, "*Measuring building occupancy using existing network infrastructure*," Green Computing Conference and Workshops (IGCC), 25-28 July 2011.

[17] T.W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse, "*Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors*," the 10th ACM Conference on Embedded Network Sensor Systems (SenSys'12),, New York, NY, USA, 2012.

[18] V.L. Erickson, M.A. Carreira-Perpinan, A.E. Cerpa, "*OBSERVE: Occupancy-based system for efficient reduction of HVAC energy*," the 10th Information Processing in Sensor Networks (IPSN'11), April 2011.

[19] F. Oldewurtel, D. Sturzenegger, M. Morari, "*Importance of occupancy information for building climate control*," Applied Energy, Volume 101, January 2013.

[20] A. Beltran, V.L. Erickson, A.E. Cerpa, "*ThermoSense: Occupancy Thermal Based Sensing for HVAC Control*," the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys'13), New York, NY, USA, 2013.

[21] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, Y. Agarwal, "*Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings*," the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys'13), New York, NY, USA, 2013.

[22] M. Gruteser, G. Schelle, A. Jain, R. Han and D. Grunwald, "*Privacy-aware location sensor networks*," HotOS IX: the 9th Workshop on Hot Topics in Operating Systems, Lihue, Hawaii, USA, May 2003.

[23] Z. Ghahramani and M.I. Jordan. *Factorial hidden Markov models*. Machine Learning, 29:245-273, 1997.

[24] P. McDaniel, S. McLaughlin, "*Security and privacy challenges in the smart grid*," Security & Privacy, IEEE, vol.7, no.3, pp.75-77, May-June 2009.

[25] S.R. Rajagopalan, L. Sankar, S. Mohajer, H.V. Poor, "*Smart meter privacy: a utility-privacy framework*," the IEEE International Conference on Smart Grid Communications (SmartGridComm), pp.190-195, 17-20 Oct, 2011.

[26] G. Pallapa, M.D. Francescoy and S.K. Das, "*Adaptive and context-aware privacy preservation schemes exploiting user interactions in pervasive environments*," the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'12), pp.1-6, June 2012.

[27] G. Pallapa, N. Roy and S. K. Das, "*A scheme for quantizing privacy in context-aware ubiquitous computing*," the 4th International IET Conference on Intelligent Environments, pp.1-8, July 2008.

[28] U. Hengartner and P. Steenkiste, "*Avoiding privacy violations caused by context-sensitive services*," Pervasive and Mobile Computing, 2(4):427-452, 2006.

[29] A. Abbasi, A. Khonsari and M. Talebi, "*Source location anonymity for sensor networks*," the 6th IEEE Conference on Consumer Communications and Networking Conference (CCNC'09), pp.588-592, 2009.

[30] B. Alomair, A. Clark, J. Cuellar and R. Poovendran, "*Towards a statistical framework for source anonymity in sensor networks*," IEEE Transactions on Mobile Computing, vol.12, no.2, pp.248-260, Feb 2013.

[31] S. Chakraborty, K.R. Raghavan, M.P. Johnson and M.B. Srivastava, "*A framework for context-aware privacy of sensor data on mobile systems*," in Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (HotMobile'13), New York, NY, USA, 2013.

[32] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin and N. Triandopoulos. "*AnonySense: privacy aware people-centric sensing*," the International Conference on Mobile Systems, Applications, and Services (MobiSys'08), pp.211-224, June 2008.

[33] B. Hoh and M. Gruteser, "*Protecting location privacy through path confusion*," the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm'05), 2005.

[34] W. He, X. Liu, H.V. Nguyen, K. Nahrstedt and T. Abdelzaher, "*PDA: Privacy-Preserving Data Aggregation for Information Collection*," ACM Trans. Sensor Networks 8, 1, Article 6, August, 2011.

[35] Y. Li and J. Ren, "*Preserving source-location privacy in wireless sensor networks*," the 6th Annual IEEE Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON'09), pp.493-501, 2009.

[36] K. Mehta, D. Liu and M. Wright, "*Location privacy in sensor networks against a global eavesdropper*," the 15th IEEE International Conference on Network Protocols (ICNP'07), pp.314-323, 2007.

[37] C. Bettini, X.S. Wang and S. Jajodia, "*Protecting privacy against location-based personal identification*," the 2nd VLDB Workshop SDM, 2005.

[38] T. Xu and Y. Cai, "*Feeling-based location privacy protection for location-based services*," the 16th ACM Conference on Computer and Communications Security (CCS'09), New York, NY, USA, 348-357.

[39] N. Li, N. Zhang, S. Das and B. Thuraisingham, "*Privacy preservation in wireless sensor networks: a state-of-the-art survey*," Elsevier Journal on Ad Hoc Networks, 7(8):1501-1514, 2009.

[40] M. Gruteser and D. Grunwald, "*Anonymous usage of location-based services through spatial and temporal cloaking*," in MobiSys, pp.31-42, New York, NY, USA, 2003.

[41] J. Krumm, "*A survey of computational location privacy*," the Personal Ubiquitous Computation, 2009.

[42] A. Beresford and F. Stajano, "*Location privacy in pervasive computing*," Pervasive Computing, IEEE, vol. 2, no. 1, pp.46-55, Jan-Mar 2003.

[43] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec and J.-P. Hubaux, "*Quantifying location privacy*," the IEEE Symposium on Security and Privacy, May 2011.

[44] M. Enev, J. Jung, L. Bo, X. Ren and T. Kohno, "*SensorSift: balancing sensor data privacy and utility in automated face understanding*," in Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC'12), New York, NY, USA, 2012.

[45] J. Petzold, "*Augsburg indoor location tracking benchmarks*," Technical Report, Institute of Computer Science, University of Augsburg, April 2004.

[46] D. Madigan, E. Einahrawy, R.P. Martin, W.-H. Ju, P. Krishnan, A.S. Krishnakumar, "*Bayesian indoor positioning systems*," the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), March 2005.

[47] J. Kolodziej, S.U. Khan, L. Wang, N. Min-Allah, S.A. Madani, N. Ghani, H. Li, "*An Application of Markov Jump Process Model for Activity-Based Indoor Mobility Prediction in Wireless Networks*," Frontiers of Information Technology (FIT), pp.51,56, 19-21 Dec. 2011.

[48] C.M. Bishop, "*Pattern recognition and machine learning*," Information Science and Statistics, Springer-Verlag New York, Inc., Secaucus, NJ, USA.