Enabling Trajectory Constraints for Usage Control Policies With Backtracking Particle Filters

Philipp Marcus, Moritz Kessel and Claudia Linnhoff-Popien
Institute of Computer Science
Ludwig Maximilian University of Munich
{philipp.marcus, moritz.kessel, linnhoff}@ifi.lmu.de

***Abstract*—A number of studies extended access control policies with constraints, aiming at the restriction of mobile users' access to appropriate authorized areas. Recent research proposed to rely on usage control instead, in order to allow for continuous checks of the user's location. A drawback of those approaches is that they rely on crisp trajectory estimates, i.e., spatio-temporal paths, not considering occurring uncertainty. This makes those approaches impractical for indoor applications, where occurring measurement errors are typically large compared to authorized areas. Thus, in this study, we propose extensions for usage control policies to constrain users to an authorized area for the duration of access. We adhere probabilistic trajectories derived from backtracking particle filters combined with WiFi fingerprinting. However, the main contribution is a risk-based model for deriving usage decisions based on risk factors instead of conventional thresholds. Our results show, that particle filters are crucial due to inaccuracy in WiFi positioning. We achieve a true- and false-positive rate of 80% and 6.7%. Finally, this allows to effectively constrain access to appropriate areas in indoor scenarios.**

***Keywords*—Mobile Usage Control; Indoor Positioning; Backtracking Particle Filter; Location-based Access Control**

I. INTRODUCTION

The significantly increasing popularity of mobile devices offers mobile access to resources from everywhere. However, this arises the inherent risk of access requests to critical resources from inappropriate areas, e.g., from outside a company's site or neighboring offices of foreign companies. To solve this problem, much study in recent years focused on location-based extensions of existing access control models, i.e., role-based access control (RBAC), mandatory access control (MAC) or discretionary access control (DAC) [1]. These extensions allow to refine access rights of mobile users with location predicates. This way, the location of users, accessed resources or both, can be constrained to certain areas or to predefined mutual relations. A drawback of those approaches is that after an access request was granted, the according rights won't be revoked when users move on to possibly inappropriate locations. As a remedy, the change to usage control mechanisms was recently discussed, which focus on the concept of controlling the usage of a resource continuously based on iterative checks [2], [3]. Location predicates applied to this model focus on constraining trajectories, i.e, the covered path of a user in the spatio-temporal space. Typically, trajectories are defined as ploy-lines and created using interpolation on crisp location measurements, for example measured with GPS. Trajectory constraints are used to constrain usage rights to users with trajectories that satisfy predefined boundary conditions. One example is to restrict the path to be contained within a single authorized area (AA), e.g., an office or room. This kind of trajectory constraint is called a containment constraint (CC) for the rest of this paper. The mentioned existing approaches for constraints on trajectories do not account for measurement uncertainty when assuming a crisp poly-line as the user's trajectory. Independently, in the research area of indoor positioning, and tracking in particular, WiFi fingerprinting in combination with backtracking particle filters (BPF), a special Bayesian filter, were shown to yield very promising results for estimating user trajectories [4]. Their performance stems from reducing the negative impact of single location measurement outliers. Additionally, BPFs allow for a probabilistic representation of trajectory estimates over probability density functions (PDF) that are sampled by a set of particles. Every particle represents a hypothesis of the user's past trajectory.

So far, techniques for coping with the probabilistic representation of trajectories in constraints for usage control policies have not been studied. This drawback even makes existing approaches impractical in indoor scenarios, where typically WiFi fingerprinting is used for positioning: Here, when creating a crisp trajectory like required by existing approaches, simply stringing together single location measurements is not sufficient, as occurring errors can easily cause the estimated trajectory to indicate a leaving of the AA erroneously. This causes unacceptable high false decision rates and impractical high risk. Furthermore, existing approaches even prevent trajectory-based usage control to benefit from the promising accuracy achieved with existing BPF and their probabilistic trajectories. Until now, in indoor environments, there exists no means to reliably constrain mobile usage of resources to predefined AAs. This might even make it impossible to obey according legal security and safety constraints in companies.

In order to facilitate CCs on probabilistic trajectories returned by BPFs, our contribution is threefold: At first, Section II presents an architecture for the continuous evaluation of CCs assigned to usage control policies. Here, also the underlying attacker model is defined. Subsequently, Section III first gives a theoretical overview on BPFs and describes how trajectory estimates are derived in our approach. Based on these results, Section IV gives a formal definition of CCs based on the probabilistic representation of trajectories in our BPF. In order to minimize CCs computational overhead, an incremental adaptation is proposed. The main contribution of this section is our risk-based approach for deriving usage decisions, which picks the decision with the lowest risk. Here, risk factors are derived based on time dependent cost functions of false-

 ISBN: 978-1-61208-313-1

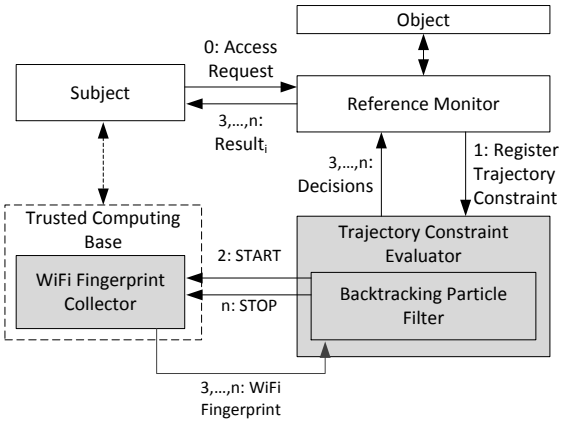

Fig. 1. The new architecture components (gray) continuously collaborate to derive and report usage decisions for the registered trajectory constraints.

negative and false-positive decisions and estimates of a CC's confidence. To the best of our knowledge, only approaches with predefined and static thresholds have been defined up to now [5], [6]. The proposed BPF and evaluation strategies for CCs are evaluated in Section V with 60 trajectories and a database consisting of 206 fingerprints covering 1400 m^2. Finally, Section VI concludes the paper.

II. Architecture and Attacker Model

Typical usage control policies incorporate attribute-based access control policies for continuous usage, e.g., UCON ABC [2]. Here, a set of usage rules states which subject is allowed to use a specific object. Decisions about revoking ongoing usage are continuously repeated. However, to the best of our knowledge, there exists only one approach up to now, that explicitly describes the extension of usage control policies with trajectory constraints [3]. Instead of probabilistic representations, only a definition on crisp trajectories is given without a possible underlying architecture. However, we define an architecture that includes components for deriving trajectory estimates and evaluating CCs: Its basic component is the *trajectory constraint evaluator* (TCE), which is able to provide trajectory-aware usage control policies and their reference monitors with Boolean decisions about the satisfaction of applied CCs. Figure 1 depicts the introduced components (gray) along with the necessary message flow. Each time an usage request arrives at the policy's reference monitor, the associated CCs for the responsible usage control rule are looked up and registered at the TCE. In the following steps, the TCE informs the mobile user's *WiFi fingerprint collector* (WFC) to start continuously providing WiFi fingerprints. The WFC is executed directly on the accessing user's mobile device and needs to be under control of a trusted computing base in order to allow for unmanipulated measurements. From now on, the WFC collects fingerprints of the received signal strength (RSS) of surrounding WiFi access points (APs) and sends the measured values along with the time-stamp of their recording digitally signed to the TCE. This needs to be continued until the mobile usage is revoked by the reference monitor or quit by the user. In order to prevent users from appointing WFCs on a foreign mobile device as their own, making trajectory evaluation useless, usage requests need to be constructed by the trusted computing base and additionally have to transmit

1: **function** BAYESIAN_FILTER($bel(x_{t-1})$, z_t)
2: **for all** x_t **do**
3: $\overline{bel}(x_{t-1}) = \int p(x_t|x_{t-1})\ bel(x_{t-1})\ dx_{t-1}$
4: $bel(x_t) = \eta^{-1}\ p(z_t|x_t)\ \overline{bel}(x_{t-1})$
5: **end for**
6: **return** $bel(x_t)$
7: **end function**

Fig. 2. General concept of Bayesian Filters [7].

the service access point of the user's own WFC to the reference monitor. For the rest of this paper, we concentrate on estimating users' trajectories within the TCE and the evaluation of the described CC based on these estimates.

The presented architecture is conform to our attacker model: An attacker is defined as a mobile user that manipulates estimates about his trajectory in order to obtain usage rights for a given resource by the reference monitor. In our case, we assume attackers that are able to *1)* manipulate sensor data of their mobile device, *2)* delay the provisioning of WiFi measurements to the TCE and, *3)* move freely within and around the AAs referenced in any CC. In contrary, he is not able to *1)* manipulate clock data or the received signal strength (RSS) in WiFi measurements, *2)* replay old WiFi measurements and, *3)* manipulate the WiFi infrastructure.

III. Estimating Trajectories with Backtracking Particle Filters

In this section, our BPF specialized for the evaluation of CCs and its theoretical foundations are presented.

A. Basic Concepts of Backtracking Particle Filters

Particle filters (PF) represent a recursive, non-parametric Bayesian filter with a discrete representation of the posterior by a set of particles of size m. In general, Bayesian filters like the particle filter allow to recursively calculate a *belief* $bel(x_t) = p(x_t|z_{1:t})$ of the system's state for a time-stamp t from already observed measurements $z_{1:t} = \langle z_1, z_2, \ldots, z_t \rangle$ [7]. Internally, the algorithm consists of two parts: The *prediction* step first computes a prior $\overline{bel}(x_t) = p(x_t|z_{1:t-1})$, called the *prediction*, before incorporating the latest measurement. This computes as the integral sum of the *prediction model*, describing the probability of getting to state x_t from a state x_{t-1} and the previous posterior $bel(x_{t-1})$. In the *update* step, the current belief is computed from the *prediction* by incorporating the measurement probability $p(z_t|x_t)$ along with a normalizing constant $\eta = p(z_t|z_{1:t-1})$. The algorithm is depicted in Figure 2. For the field of localization and tracking, PFs implement this algorithm by sampling the posterior probability density function $bel(x_t)$ with a set of m particles $\mathcal{X}_t = \left\langle x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[m]} \right\rangle$, each one representing a concrete instantiation of the state with information about the position, the velocity and the orientation. The *prediction step* is implemented by applying a mobility model to each particle that predicts its new position, orientation, and velocity. This leads to a updated particle set $\overline{\mathcal{X}}_t$ that approximates $\overline{bel}$. The *update step* is implemented by importance re-sampling: Each particle $x_t^{[i]}$ is assigned a weight $w_t^{[i]}$, called its importance factor, according to the measurement z_t. In order to get a updated set

of particles $\mathcal{X}_t$ that is approximately distributed to $bel(x_t)$, m particles are drawn with replacement from the set $\overline{\mathcal{X}}_t$. Here, the probability of drawing a particle is proportional to its weight.

B. Deriving Measurements and Measurement Probabilities

In our system, single location measurements z_t are computed by WiFi fingerprinting: When the user's mobile device measures a RSS for a set of APs, the most likely position is determined from a previously recorded fingerprint database. It computes as the weighted center of mass of fingerprints selected by the k-nearest-neighbors algorithm [8]. Given a location measurement z_t, we derive the measurement probability $p(z_t|x_t^{[i]})$ from a bi-variate Gaussian pdf that describes the error distribution, based on our previous work [8]. When recording the fingerprint database, not only the area of modeled AAs should be covered. Otherwise, as the range of APs must be assumed to be larger than the modeled AAs, a potential attacker could simply stand outside the AA and the positioning algorithm will have no choice but choosing fingerprints as k-nearest-neighbors that were recorded within the AA. This allows an attacker to obtain a position fix that indicates a position within the AA, possibly leading the usage control policy to a false-positive. To solve this problem, we propose to determine the set of those APs that are receivable within the authorized region in at least one point. Next, the union of the coverage areas of this set of APs needs to be covered when recording the fingerprint database. In general, this increases the effort of recording the fingerprint database. For unbiased positioning, the spatial density of the fingerprint database should be uniform throughout the covered site.

In order to annul any of an attackers sensor manipulations, the underlying WiFi fingerprinting and PF algorithm must not employ sensor data. In particular, compass data has been shown to have a positive effect on positioning accuracy to reduce influence of blocking effects of the human body on RSS values [8]. Hence, it is not possible to apply the corresponding technique of recording each fingerprint once for each cardinal direction and choose only those as kNN that have been recorded with the user's current orientation. Therefore, in order to prevent the possibility of an intentional attack, our system needs to accept a possibly lower positioning accuracy.

C. The Application of Backtracking for Refining Estimates

During the *update step* via re-sampling, typically some particles from $\overline{\mathcal{X}}_t$ are not contained in $\mathcal{X}_t$ due to their low weight and are said to *die* whereas others might be drawn a multiple times. In such cases, with each particle, also its assigned hypothesis about a possible user trajectory dies. This allows to refine the knowledge about possible user trajectories up to time t by discarding trajectories associated to dead particles via *backtracking*, which represents a BPF [4]. In BPFs, the knowledge about the past is only refined by future *update steps* as single particles with their assigned estimated trajectory can only be discarded and not newly created. In traditional tracking or positioning systems, for each point in time a single position is computed as the mean value of all particles, leading to one estimated trajectory. This way, information about single existing hypotheses is lost and the single trajectory computed from the means might completely satisfy a CC though none of the estimated trajectories does so. Therefore, trajectory constraints should consult the whole set of trajectories in order to exploit all available information to finally derive confidence values.

D. The Prediction Step: Deriving Trajectory Estimates

To allow for a detailed representation of possible trajectories, we realize the *prediction step* by partitioning the time-span between two *update steps* in sub-intervals of maximally 800 ms, which corresponds to the typical time human users need to take a step. After each passed sub-interval, each particle's trajectory is expanded with a segment. The segments represent a possible movement of the user in this sub-interval. Here, a single segment is constructed as a line by assuming a linear movement from its last position $\mathbf{l}$ with a velocity v in a direction α for the duration of the current sub-interval. Let *Loc* be an arbitrary polygon representing the AA of a CC, we write $\tau(x_t^{[i]})$ *within Loc* iff the segments assigned with $\tau(x_t^{[i]})$ are completely contained within *Loc*. Let $t = 0$ denote the point of time when the usage right was granted and the BPF initialized. Appending the trajectory estimates from all past state estimates a particle $x_t^{[i]}$ was generated from, an estimate for the user's trajectory since $t = 0$ is derived and denoted as $\tau(x_0^{[i]}) \circ \tau(x_1^{[i]}) \circ \ldots \circ \tau(x_t^{[i]})$. This will serve as a key deciding factor for our trajectory-based usage control mechanism.

When constructing a particle's $\tau(x_t^{[i]})$, single segments should be created by a mobility model that is appropriate for the application to usage control in terms of the security implications of our attacker model: Similar to plain WiFi fingerprinting, BPF algorithms were shown to achieve higher accuracy by the application of inertial sensor data, too, and especially benefit from step detection algorithms [9]. This data can be used to perform the *prediction step* based on dead reckoning with the measured sensor values. However, all existing approaches assume benevolent users that do not fake steps by shaking a smart-phone or pretend other movements. If directly applied to our scenario, an attacker could fool the system to assume trajectories that do not leave a prescribed AA though the attacker has left it. Again, it is crucial to accept possibly lower positioning accuracy in order to hamper attacks. Therefore, particles are modeled to follow a random waypoint mobility model, which appends segments to $\tau(x_t^{[i]})$ of each particle $x_t^{[i]}$. One possibility is the model presented by Widyawan et al. [4]. This allows to model linear movement according to the boundary conditions of humans, based on the angle and velocity of the preceding segment. However, basically, any mobility model that is independent of a mobile device's inertial sensor data and based on map matching can be applied. This way, each particle is assigned a hypothesis of the trajectory the user could have walked since the *update step* at time t until the subsequent *update step* is performed.

In order to obtain realistic estimates for trajectories, we also apply the technique of map matching [4], [9]. As for each particle the plausible choices for its next segment are limited by the characteristics of the underlying building plan, we require that particles must not cross walls. During the construction of $\tau(x_t^{[i]})$, this is realized by retrying to infer a valid succeeding segment that is a realistic extension of the trajectory and does not cross walls, until a predefined threshold of maximum tries is exceeded. In such cases, the weight $w_t^{[i]}$

 ISBN: 978-1-61208-313-1

of the particle is 0, despite the probability that might arise from the latest measurement z_t. Hence, the particles' weights are influenced by the *prediction step* and are set to 0 if only wall-crossing segments could be derived within a maximum amount of retries:

$$w_t^{[i]} = \begin{cases} 0, & \text{no valid } \tau(x_t^{[i]}) \text{ found} \\ p(z_t|x_t^{[i]}), & \text{otherwise} \end{cases} \quad (1)$$

Trajectories constructed this way are robust against sensor manipulations of attackers and can finally be supplied for the evaluation of a CC in order to derive usage decisions.

IV. Containment Constraints for Backtracking Particle Filters

The trajectories computed by the adapted BPF are applied to evaluating CCs. In this section, we first discuss the differences to existing approaches and define the concept of CCs formally. Based on this result, an incremental algorithm for their evaluation is presented. Finally, we present a risk-based model for deriving usage decisions based on the current confidence of an evaluated CC.

A. Applicability of Traditional Trajectory Constraints

When using classical location providers like GPS for sampling a user's trajectory, a sample consists of a crisp location and a time-stamp. In the field of moving object databases (MOD), this has been used to define beads, i.e., ellipsoid geometries that contain all points that could be visited during the collection of two samples under the assumption of a maximum velocity [10]. Trajectories are hence affected with uncertainty and described as a sequence of beads. The real trajectory is completely contained within the given beads. With each new sample arriving at a MOD server, the current sequence of beads is extended by a new element. This allows a clear distinction of past and future trajectories. In such cases, trajectory-based usage control can be realized using traditional spatio-temporal queries, assessing to what degree the given beads satisfy the containment within a room. However, as mentioned above, BPFs showed much higher accuracy in indoor scenarios and consequently our trajectory estimates are derived using this method. Those don't form beads and thus classical spatio-temporal queries can't be used directly as an implementation of CCs. Furthermore, re-sampling prevents the clear distinction between past and future trajectories. Even the known representation of beads is not possible without an additional density estimation based on particles' trajectories. These differences of trajectory estimates of BPFs compared to sequentially constructed ones in traditional MODs need to be respected here. Hence, in the following paragraph, we present the formal definition of CCs for usage control policies.

B. Containment Constraints for Backtracking Particle Filters

A *containment constraint cc* is defined as a function mapping from an authorized area *Loc* and the current set of particles $\mathcal{X}_t$ to a confidence value denoting the percentage of trajectory estimates that are completely contained within *Loc* since the usage right was granted at $t = 0$:

Definition 1 (Containment Constraint)
The containment constraint (cc) on a set of particles and an authorized area is defined as:

$$cc\,(\mathcal{X}_t, Loc) = |\mathcal{X}_t|^{-1} \cdot \left| \left\{ x_t^{[i]} \in \mathcal{X}_t \mid \forall k \in \{0, \ldots, t\} : \tau(x_k^{[i]}) \textit{ within } Loc \right\} \right| \quad (2)$$

with $\mathcal{X}_t$ being the current set of particles since the last update step at time t and Loc being a polygon in $\mathbb{R}^2$ representing the authorized area.

Basically, this constraint needs to be evaluated after every expansion of particles' trajectories during the *prediction* and after each *update step*, as both can influence the percentage of trajectories that satisfy the constraint. Note, that Definition 1 is an adaptation of the well studied *possibly always* spatial query [10]. However, when evaluating CCs, the most computationally demanding step is to check $\tau(x_k^{[i]})$ *within Loc* for each particle $x_k^{[i]}$, for each time-span in history. Employing the discussed properties of our BPF, we define a more efficient, incremental evaluation of CCs: By assigning and incrementally updating a Boolean *is_valid* to each particle in $\mathcal{X}$ it is possible to highly reduce the required number of these checks. This Boolean describes if the trajectory assigned to a particle $x_t^{[i]}$ satisfies or violates the CC that is currently under evaluation. As each single expansion might cause a trajectory to violate the *cc*, in each of its re-evaluations also the property *is_valid* needs to be updated. As single trajectories can't recover from a violation already detected in prior evaluations, the *is_valid* property only needs to be updated for particles with *is_valid* = *True*. The confidence can then easily be computed by counting the percentage of particles that still satisfy the constraint, as depicted in Figure 3. Obviously, this implements Definition 1,

```
1: function INCREMENTAL_CC( X_t, Loc )
2:     for all x_t ∈ X_t do
3:         if x_t.is_valid ∧ ¬ τ(x_t) within Loc then
4:             x_t.is_valid ← False
5:         end if
6:     end for
7:     return |X_t|^-1 · |{x_t ∈ X | x_t.is_valid}|
8: end function
```

Fig. 3. The algorithm for incrementally computing confidence values.

as $x_0^{[i]}.is\_valid = \tau(x_0^{[i]})$ *within Loc* and,

$$x_{t+1}^{[i]}.is\_valid \;= x_t^{[i]}.is\_valid \wedge \tau(x_{t+1}^{[i]}) \textit{ within } Loc \quad (3)$$

Thus, in contrast to Definition 1, *incremental_cc* only needs to evaluate $\tau(x_{t+1}^{[i]})$ *within Loc* for the current τ and only for particles with $x_t^{[i]}.is\_valid = True$, i.e., those that did not already violate the constraint in any prior evaluation.

C. Deriving Risk-Based Usage Control Decisions

Based on the confidence values returned by the policy's underlying CC, usage control decisions need to be derived after each *update step* and continuously during each *prediction step*. In detail, the two usage decisions comprise either to revoke the usage right or keep on granting it. We choose that decision with

the lowest risk if it was wrong w.r.t. the ground truth. Here, we model risk as the product of probability and costs. Each of these decisions brings costs [11]. The costs for a correct decision, i.e., a true-positive or a true-negative are assumed to be 0. The modeling of costs for false decisions needs to respect that an attacker might retain new WiFi measurements in order to elongate the *prediction step* by an arbitrary time to possibly prevent a revocation of his usage right. Hence, costs of a false-positive are assumed to increase with ongoing time of a *prediction step*, as the occurring damage might increase, i.e., by dumping protected data. Its costs are modeled by a monotonically increasing function $c_{fp}(t) = f(t)$. This also allows to express the intuitive notion of revoking the usage right after a predefined time-off. However, the costs of a false-negative are modeled as $c_{fn}(t) = const$ as we assume the costs that occur from refused usage to be constant with time.

To derive a usage decision directly after the *update step*, first the confidence (c) of the CC is computed. As the *prediction step* has not yet started, here $t = 0$ and the usage right is revoked if the risk of a false-positive exceeds the risk of a false-negative, i.e., $c_{fp}(0) \cdot (1 - c) \geq c_{fn}(0) \cdot c$. However, to derive usage decisions during the *prediction step*, we estimate the maximum risk for both decisions for any point of time t during the *prediction step*: The maximum risk of keep granting is directly influenced by the lowest possible confidence about the CC at time t. Note, that the observed confidence of the randomly moved particles typically will be higher than this lower bound. Contrary, the maximum risk of revoking the usage at time t is proportional to the highest probability that the user could still be within the AA and thus to its initial confidence. In line with existing risk-based approaches to access or usage control [11], [12], again we revoke usage as soon as the maximum risk of a revoke is lower than the maximum risk of a grant. The confidence values used for computing the maximum risk factors are derived from the CC's initial confidence directly at the begin of the *prediction step*: The lowest possible confidence $p_{max\_out}(t)$ of the CC at any point t in the *prediction step* can be computed by assuming all particles that satisfied the CC at begin of the *prediction step* to leave the AA on the shortest path. The highest possible confidence $p_{max\_in}(t)$ occurs when all particles that satisfied the CC at begin of the *prediction step* stay within the AA and thus still satisfy the CC throughout the *prediction step*. This value is a constant. Consequently, the corresponding maximum risk values for false-positive and false-negative decisions compute as $c_{fp}(t) \cdot p_{max\_out}(t)$ and $c_{fn}(t) \cdot p_{max\_in}(t)$ respectively. Directly when a *prediction step* starts, this allows to precompute a time-stamp t_{revoke}, representing the point in time when the risks of a false-positive are too high and thus when the revoke should be issued. The advantage of this approach is that our system doesn't need to evaluate the CC during the *prediction step* and is independent from the number of randomly moved particles that represent an attacker with their movement. Furthermore, this way the maximum time until the next update depends on the confidence of the CC at the beginning of a *prediction step* and revocations can be derived based on occurring risk factors.

V. Evaluation

In this section, the advantages of the proposed approach are evaluated in a comprehensive test set.

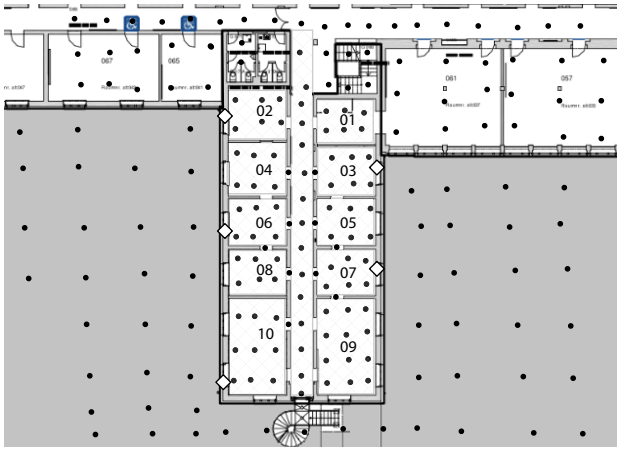

Fig. 4. Reference positions of our WiFi fingerprint database (black dots) and installed APs (diamonds). AAs were defined using the labelled rooms; outdoor areas are drawn in dark gray.

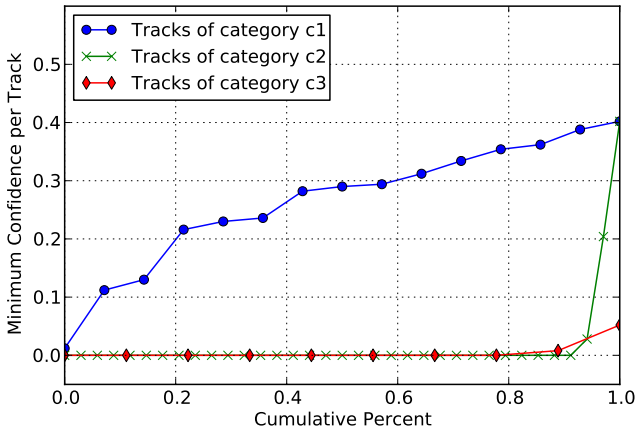

Fig. 5. Minimum observed confidence for each trajectory category.

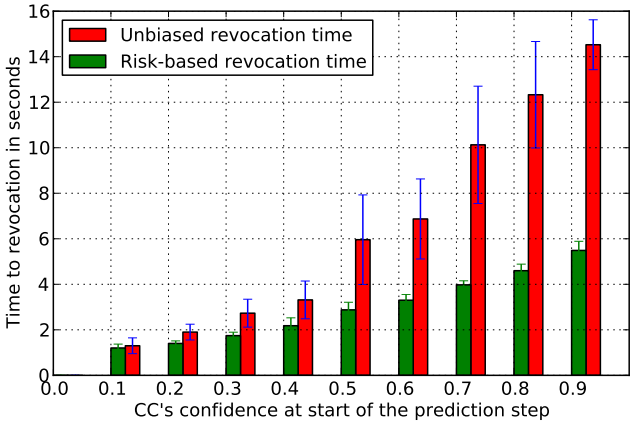

Fig. 6. Revocation times in dependence on the CC's confidence at begin of a *prediction step* for random movement and for the derivation of t_{revoke}.

A. Test Environment and Hardware

For evaluating the approach, we recorded a fingerprint database consisting of 206 WiFi fingerprints, each computed as the mean value of single 20 measurements with 4 for

every cardinal direction. We installed 5 APs and employed their RSS values for positioning. The single WiFi fingerprints were collected using a HTC Desire smart-phone. The covered site comprehends about 1400 m^2 and is depicted in Figure 4. Additionally, we defined the following 5 AAs on the rooms contained in the hatched area of Figure 4, here referenced by their depicted identifiers: (01), $(06;08)$, $(05;07)$, $(07;09)$ and $(01–06)$ along with the part of the floor in-between. For each of these areas, 12 possible user trajectories were recorded, each approximately 60 s long and consisting of a sequence of observed RSS values recorded at least every 1.5 s. Each trajectory's ground truth was supplied manually. For each modeled AA, the recorded trajectories can be classified as follows: Three trajectories that run completely within their AA, one with a user standing still inside. Three trajectories that run outside but near to the AA and inside the building, one with a user standing still. Two trajectories that leave and re-enter the AA, and three trajectories that run near the AA outside the building. The recorded trajectories represent three categories w.r.t. their ground truth: satisfying the AA all time (*c1*), violating it from begin on (*c2*), and satisfying the AA at begin but violating it later on (*c3*).

B. Evaluation Results

In order to detect trajectories that satisfy or violate a CC, the minimum observed confidence for a given trajectory should correspond to the trajectory's category. Hence, for each of the three categories *c1*-*c3*, the minimally observed confidence of assigned trajectories was computed and is depicted as a cumulative distribution function (cdf) in Figure 5. Clearly, in over 80% of all cases, the minimum observed confidence for trajectories of category *c1* was greater than 20%. In contrast, the trajectories of *c2* and *c3* showed a minimum confidence of 0% in over 90% of all cases. We considered these proportions in the definition of cost functions according to Section IV-C by choosing the ratio $c_{fp}(0)/c_{fn}(0) = (100\% - 20\%)/20\% = 4/1$. c_{fp} rises compliant with the sampling rate of measurements. The classification results based on our cost functions are compared to those of a single crisp trajectory derived from stringing together observed location measurements, conform to existing approaches to trajectory-based usage control [3]. Table I depicts the results. As the results indicate, the crisp trajectory

TABLE I. CLASSIFICATION RESULTS.

Used Approach	TP	FP	TN	FN
Our BPF	80%	6.7%	93.3%	20%
Stringed location measurements	13.3%	0.0%	100%	86.7%

has a true-positive (TP) rate of 13.3%, which makes its application nearly impractical. However, our approach yielded a TP rate of 80%. The crisp trajectory showed a true-negative rate (TN) of 100% in contrast to our approach, which showed a slightly lower TN rate of 93.3%. Consequently, our approach shows higher false-positive (FP) classifications and slightly higher chances of a misuse but outperforms the crisp trajectory with its TP rate by far, which results in a far higher availability of the usage right if really justified. To assess the benefits of the proposed *incremental_cc* evaluation, we determined the mean count of pruned particles for trajectories of each category. For trajectories in *c1*, *incremental_cc* could prune 22% of all particles in the mean in contrast to 94% for category *c2* and 72% for category *c3*. The outcomes show the strong correlation to the number of violating particles and indicate that the incremental evaluation is expected to prune at least 22% of all particles by mean. Finally, we compare the impact of deriving risks in the *prediction step* from the lowest possible confidence for any point in time instead of adhering the iteratively updated confidence deduced from the particles' random movement. The results are depicted in Figure 6. Clearly, the proposed model of assuming the lowest possible confidence yields more realistic revocation times, as in the other approach, particles often get stuck within certain rooms when following a mobility model with random movement.

VI. CONCLUSION

In order to enable usage control policies to benefit from trajectory constraints in indoor scenarios, we proposed back-tracking particle filters (BPF) to derive probabilistic trajectory estimates and discussed requirements to complicate attacks. Subsequently, we proposed the concept of containment constraints (CC), which require a user to stay within a certain authorized area (AA) for the duration of his usage of a protected resource. An improved evaluation strategy based on the discrete and probabilistic representation of potential trajectories was presented. In order to allow for a comprehensible revocation of usage rights we proposed to compute risk factors for a false-positive and a false-negative, choosing that decision with the lowest risk. In contrast to existing research, our approach respects occurring uncertainty of trajectory estimates and works on according probabilistic representations. This allows to exploit all available information when deriving usage decisions. Furthermore, to the best of our knowledge, no approach for deriving appropriate revocation times has been proposed before. Thus, the main contribution of this work is a mechanism for enforcing CCs in usage control policies based on probabilistic trajectories represented by particles, constructed by a BPF. In the evaluated indoor scenario, this concept shows a very encouraging true-positive rate of 80% at the price of a false-positive rate of 6.7%. However, for crisp trajectories created by stringing together single location measurements, like required by all existing approaches, only an impractical true-positive rate of 13.3% could be observed. Finally, our approach allows for a robust enforcement of CCs in indoor scenarios and to constrain mobile usage of resources to suitable areas and rooms without high extra expenses for additional positioning infrastructure. Future work should focus on integrating CCs in policies, hampering location spoofing with WiFi fingerprinting and using appropriate implicit authentication methods to couple user and device locations.

ACKNOWLEDGMENTS

Thanks to Sebastian A. Amft for helping with the test data.

REFERENCES

[1] E. Bertino and M. S. Kirkpatrick, "Location-based access control systems for mobile users: Concepts and research directions," in *Proceedings of the 4th ACM SIGSPATIAL Int'l Workshop on Security and Privacy in GIS and LBS*. ACM, 2011, pp. 49–52.

[2] J. Park and R. Sandhu, "The UCON_{abc} usage control model," vol. 7, no. 1. ACM, 2004, pp. 128–174.

 ISBN: 978-1-61208-313-1

[3] M. L. Damiani, E. Bertino, and C. Silvestri, "Approach to supporting continuity of usage in location-based access control," in *12th IEEE Int'l Workshop on Future Trends of Distributed Computing Systems*. IEEE, 2008, pp. 199–205.

[4] Widyawan, M. Klepal, and S. Beauregard, "A novel backtracking particle filter for pattern matching indoor localization," in *Proceedings of the first ACM Int'l Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*. ACM, 2008, pp. 79–84.

[5] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Supporting location-based conditions in access control policies," in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*. ACM, 2006, pp. 212–222.

[6] H. Shin and V. Atluri, "Spatiotemporal access control enforcement under uncertain location estimates," in *Data and Applications Security XXIII*, ser. LNCS. Springer, 2009, vol. 5645, pp. 159–174.

[7] S. Thrun, W. Burgard, D. Fox *et al.*, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.

[8] P. Marcus, M. Kessel, and M. Werner, "Dynamic nearest neighbors and online error estimation for smartpos," *Int'l Journal On Advances in Internet Technology*, vol. 6, no. 1&2, 2013.

[9] M. Kessel and M. Werner, "Automated wlan calibration with a back-tracking particle filter," in *2012 Int'l Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2012, pp. 1–10.

[10] G. Trajcevski, "Uncertainty in spatial trajectories," in *Computing with Spatial Trajectories*. Springer, 2011, pp. 63–107.

[11] L. Chen and J. Crampton, "Risk-aware role-based access control," in *Security and Trust Management*, ser. LNCS. Springer, 2012, vol. 7170, pp. 140–156.

[12] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*. ACM, 2012, pp. 157–168.

 ISBN: 978-1-61208-313-1