2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress

Fine-grained Delta Privacy Preservation for Hierarchical Contexts

Xue Jiang, Yu Huang, Hengfeng Wei
State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{*xuejiang1225, hengxin0912*}*@gmail.com, yuhuang@nju.edu.cn*

***Abstract*—Rich contexts enable the provision of context-aware services on mobile smart phones, but they also introduce threats of privacy leakages. It is widely held that the key to privacy preservation is to achieve efficient tradeoffs between the utilization of contexts and the preservation of privacy. Götz et al. present δ-privacy that exploits temporal relation to decide the release/suppression and this technique achieves a good tradeoff. However, δ-privacy does not consider the hierarchy of different layers of contexts. In this paper, we propose FDH which provides fine-grained suppression over the context hierarchy and integrates δ-privacy. Our experiments on real smart phone context traces show that FDH can release a few suppression contexts in δ-privacy with limited cost and obtain relatively high utility while preserving strong privacy.**

***Keywords*-privacy preservation; mobile application;**

I. Introduction

Mobile phones today are equipped with a variety of sensors like GPS, accelerometer and gyroscope[1]. Mobile applications can thus benefit from the rich context information they are able to obtain, and provide context-aware services to the user. For example, being aware of a user's location, the mobile application can intelligently recommend restaurants or help find taxies nearby[2], [3].

While context-awareness greatly increases the quality of services of mobile applications, it also brings severe threats of privacy leakages[4]. It has been reported that in location-based applications, the risks of sharing location information may outweigh the benefits of enjoying location-based services[5].

It is widely held that the key to privacy preservation is to achieve efficient tradeoffs between the utilization of contexts and the preservation of privacy. In order to achieve such efficient tradeoffs, the notion of δ-*privacy* is proposed, which enables the user to decide in an online fashion whether to release or suppress the current context by exploiting temporal correlations among contexts[6]. For example, the user can decide to release that he is in the office, while being not willing to release that he is in the hospital. A user can also flexibly select a balance point between the privacy and utility tradeoff in order to meet the demand of privacy protection or the quality of the application. Preservation of δ-privacy guards against not only the leakage attacks from adversaries knowing the suppression system but also the inference attacks from adversaries knowing the temporal correlations among contexts.

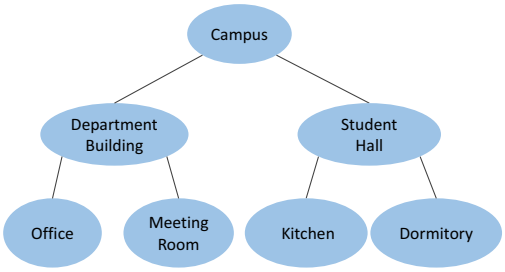

Figure 1. Location hierarchy of a campus

Though δ-privacy achieves efficient control of sensitive contexts, it does not consider the intrinsic and naturally-available *hierarchical* relation among contexts[7]. Take the location contexts of a university student as an example. The location contexts naturally have the hierarchical structure, as shown in Figure 1. In the lowest layer, we can obtain location contexts such as "in the office" and "in the meeting room". These two locations belong to the upper layer context "in the department building". The student may also be "in the dormitory" or "in the kitchen", and these two locations belong to the upper layer context "in the student hall". The upper layer contexts belong to the top level context in this location hierarchy - "in the campus".

Though δ-privacy can efficiently decide whether to release a context, the natural hierarchy among contexts can be utilized to achieve more fine-grained control. In the example above, the student may consider that the context "in the dormitory" is sensitive and want to suppress it. However, he may still be willing to share the more "vague" context "in the student hall", in order to obtain better location-based services in the student hall.

The discussions above motivate the privacy preservation scheme of *Fine-grained Delta privacy for Hierarchical contexts* (FDH). FDH achieves fine-grained control of contexts by considering the hierarchy of contexts. More specifically, when one specific piece of context is decided to be suppressed, FDH further considers its "parent context" in the context hierarchy. For each specific piece of context, FDH decides the release/suppression of it based on δ-privacy.

The FDH scheme consists of two essential parts: training and checking. FDH first trains temporal correlations and

978-1-5090-2771-2/16 $31.00 © 2016 IEEE
DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.17

suppression probabilities of contexts in each layer of the hierarchy. Temporal correlations of contexts in one layer are modeled by a Markov chain[8], [9]. The suppression probabilities of contexts in each layer compose a suppression vector and FDH uses a greedy heuristic to search the most suitable suppression vector. FDH then uses the probabilistic checking and simulatable checking to decide the suppression of hierarchical contexts. The probabilistic checking utilizes suppression vectors of the hierarchy to make decisions in constant time, while the simulatable checking utilizes temporal correlations among contexts to make decisions. The checking starts from the lowest layer to the highest layer at runtime to decide to which level in the hierarchy the context should be released.

We conduct experiments to evaluate FDH on both a PC and a smart phone, with real location traces from 106 users over the course of nine months, representing user contexts over 266,000 hours. The evaluation results demonstrate that for at least 10% of the contexts which are suppressed by δ-privacy, FDH can achieve more fine-grained privacy control and the checking cost is limited. The evaluation results also demonstrate that FDH can still obtain relatively high utility while providing fine-grained privacy control for hierarchical contexts.

II. Problem Statement

A. Overview of FDH

We assume a system that models a smart phone running various context-aware applications. It can get user contexts $x_1, x_2, ...$ periodically at discrete points in time. These contexts are the most fine-grained. We consider contexts of discrete time to avoid high energy consumption for continuously running applications[10], [11]. FDH produces a privacy-preserving output o_t for each user context input x_t. Context-aware applications cannot get raw sensor data. However, they can get the whole contexts produced by FDH.

FDH executes a hybrid privacy check to produce output contexts. The privacy checking procedure helps the system decide release or suppression. The output o_t of x_t can be three possibilities: 1) the user context x_t; 2) the "ancestor context" of x_t in the context hierarchy but not x_t itself; 3) suppression symbol $\perp$. This restriction of output reflects the standard access mechanisms in existing phones and the way many location-based mobile applications operate[5].

B. Model of user's background knowledge

Many kinds of contexts have intrinsic and naturally-available hierarchical structure[7]. For example, we can construct a context hierarchy with three layers according to location contexts of a university student mentioned in the section above. The contexts of two adjacent layers may have "parent-child" relations. The contexts can also have "ancestor-descendent" relations. For example, we regard "department building" as the parent context of "office", "office" as the child context of "department building", "campus" as the ancestor context of "office" and "office" as the descendant context of "campus". Contexts in the same layer have the same granularity, and can be modeled by a Markov chain[8], [9].

The Markov chain M_l of the layer l includes states and transition probabilities, where state denotes a user's context at a given time and transition probability $a^t_{l,i,j}$ indicates the probability of a user being in context $c_{l,j}$ at time $t+1$ given that he is in context $c_{l,i}$ at time t. The states in M_l are labeled with contexts $\{c_{l,1}, c_{l,2}, ..., c_{l,n}\}$. The states and the transition probabilities describe the frequencies of the user's contexts and temporal correlations respectively. We consider the whole contexts in the hierarchy are in a single day, so each layer l has a Markov chain M_l over a day. Each Markov chain has a "start" state and an "end" state, the times of which are 0 and T respectively. We use $\{X_{l,1}, X_{l,2}, ..., X_{l,t}\}$ to denote the generated random variables from the Markov chain M_l where each $X_{l,i}$ assumes the values from $\{c_{l,1}, c_{l,2}, ..., c_{l,n}\}$.

C. The power of adversaries

In our system, we consider adversaries which are strong enough to get the background information of each user. That is they know the Markov chain of each level in the hierarchy and the generating output context sequences of the system S. In the following, we assume that the adversaries apply Bayesian reasoning. The prior probability of the user in context $c_{l,i}$ at time t, denoted by $\Pr[X_{l,t} = c_{l,i}]$, can be computed with the Markov chain. The value of $\Pr[X_{l,t} = c_{l,i} | S(\vec{x_l}) = \vec{o_l}]$, representing the posterior probability based on the observing output sequence $\vec{o_l}$ generated at level l by the system, can be computed by following the forward-backward algorithm of a Hidden Markov Model[12], [13], [14].

D. Privacy preservation over hierarchy

In our system, privacy is defined according to a set of sensitive contexts specified by the user. Note that these sensitive contexts are the most fine-grained. Consider a user who thinks a subset SC of the whole most fine-grained contexts as sensitive. FDH considers each context of SC and its ancestor contexts as sensitive. We use s_l to denote each sensitive context of layer l. The goal of the fine-grained privacy preservation is to prevent the adversary from learning more about the user being in a sensitive context with the released sequence. Here we utilize δ-$privacy$ to describe privacy preservation for context hierarchy. For each layer l of the context hierarchy, for all possible inputs $\vec{x_l}$ sampled by the Markov chain M_l of the system, for all possible outputs $\vec{o_l}$, for all times t and all sensitive context s_l in this layer, the following inequality must be satisfied. The inequality of preserving δ-$privacy$ of a given layer l is

$$\Pr[X_{l,t} = s_l | \vec{o_l}] - \Pr[X_{l,t} = s_l] \leq \delta$$

In the context hierarchy, we expect to release the context as more fine-grained as possible, while preserving privacy. If the context cannot be released as the most fine-grained context in the lowest layer, FDH considers its ancestor contexts from the second lowest layer to the highest layer, until it finds an ancestor context which can be released. If no context can be released, FDH outputs the symbol $\perp$.

E. Utility of released contexts

The more contexts the system releases, the better the quality of the context-aware services is. So FDH tries to release as more contexts as possible, while preserving privacy. We measure the utility of the system as the weighted sum of the released contexts where each context is weighted according to the layer it belongs to. The released context which is more fine-grained has a bigger weight.

III. Fine-grained Delta Privacy Preservation

The core of the system is the hybrid privacy check, which decides whether to release appropriate context or suppress it for each context. Before processing the hybrid privacy check, we need to do some prior work including constructing context hierarchy for the most fine-grained contexts and training Markov chains based on the context hierarchy and suppression probability of each context in the context hierarchy.

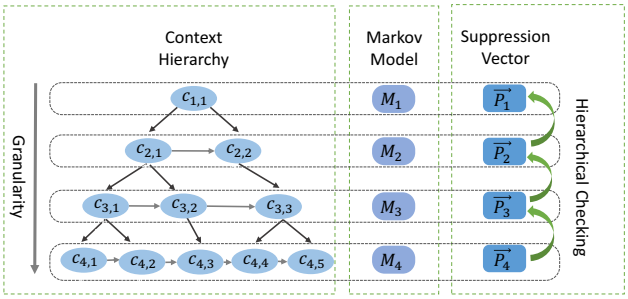

Figure 2. Main parts of the system

A. Context hierarchy

Our system can get a user's historical context traces for a long time. These historical contexts collect a user's activities and other living patterns. These contexts are the most fine-grained. Based on the knowledge about the relations among the contexts, we can construct the hierarchical structure for them. For example, in Figure 2, $c_{4,1}, c_{4,2}, c_{4,3}, c_{4,4}, c_{4,5}$ are the most fine-grained contexts, $c_{3,1}$ represents the set $\{c_{4,1}, c_{4,2}\}$, $c_{3,2}$ represents the set $\{c_{4,3}\}$, $c_{3,3}$ represents the set $\{c_{4,4}, c_{4,5}\}$, $c_{2,1}$ represents the set $\{c_{3,1}, c_{3,2}\}$, $c_{2,2}$ represents the set $\{c_{3,3}\}$, and $c_{1,1}$ represents the set $\{c_{2,1}, c_{2,2}\}$. There are four layers in the hierarchy, contexts in different layers have different granularity; the contexts in lower layer are more fine-grained. The pointer with horizontal direction indicates the correlation between the contexts of the same granularity, and the pointer with virtual direction indicates the "parent-child" correlation between contexts of different granularity.

B. Training

It is mentioned before that we assume adversaries are strong which know the background knowledge about the user. A Markov chain can provide such background knowledge of a user. In addition, we need to get the suppression probability with which each state is suppressed. These suppression probabilities will be utilized in the hybrid check. The whole suppression probabilities of each layer compose a suppression vector. So FDH should train Markov chains and suppression vectors.

User's background knowledge: A Markov chain captures frequencies of contexts and temporal correlations among them. In the hierarchy, each layer contains contexts of the same granularity. FDH exhibits a Markov chain for each layer of the hierarchy. Baum-Welch algorithm[15] is usually used to learn the parameters of a Markov chain. However, if the state path of each training sequence is achieved, the parameters can be simply learned in the following way[16]. We can estimate the transition probability $a^t_{l,i,j}$ as the times the user went from $c_{l,i}$ at time t to $c_{l,j}$ at time $t+1$ divided by the sum of times the user went from $c_{l,i}$ at time t to each possible context at time $t+1$. The pointer with horizontal direction in the first part of Figure 2 contains the transition information.

Suppression probabilities: Over the hierarchy context model, the suppression probabilities of contexts in one layer constitute a vector $\vec{P_l}$. We can get several suppression vectors over the hierarchy context model. For example, we can get four suppression vectors $\vec{P_1}, \vec{P_2}, \vec{P_3}, \vec{P_4}$ in the third part of Figure 2. For a suppression vector $\vec{P_l}$, this layer must preserve δ-*privacy* with it and the utility of it is highest among all vectors of this layer which preserve privacy. Suppose we have the suppression probability of each context, then we can simply get the emission probability of each context by one minus the suppression probability. If we use $p^t_{l,i}$ to denote the suppression probability specified for each context $c_{l,i}$ at time t, $1 - p^t_{l,i}$ denotes the emission probability.

In the context hierarchy, released contexts of different layers make different contributions to the utility; the released more fine-grained context makes bigger contribution. We assign a factor α_l to each layer which indicates the contribution of each released context of this layer. We measure the utility of a specific layer l as the expected number of released contexts with the suppression vector of this layer:

$$\text{utility} = \sum_{t\in[T], i\in[n]} \alpha_l \Pr[X_t = c_{l,i}](1 - p^t_{c_{l,i}})$$

For each layer, we need to search the most suitable suppression vector with which the utility is highest. First, we find all possible suppression vectors preserving δ-$privacy$ and then choose the vector with the maximum utility. Each suppression probability in a suppression vector can only be an value of a set $\{0, 1/d, 2/d, ..., d/d\}$ where d is a parameter specified by the user. For one suppression vector $\vec{p}$ which preserves δ-$privacy$, if another vector $\vec{q}$ dominates it, $\vec{q}$ also preserves δ-$privacy$[6]. This monotonicity property tells us if we increase the suppression probabilities we can improve privacy. Furthermore, utility has anti-monotone property, that is to say if we increase the suppression probabilities, the utility will decrease[6]. The monotonicity property of privacy and the anti-monotone property of utility allows us to adapt existing efficient greedy search algorithm to search the vector. The suppression vector is initialed to be $(1, 1, ..., 1)$. We then gradually reduce it to get a minimal vector which still preserves δ-privacy. We use algorithm ALGPR[17] to search the most suitable vector of each layer which solves the following optimization problem:

$$arg_{\vec{p_l}} \max\ utility(\vec{p_l}) \text{ subject to } isPrivate(\vec{p_l}) = true$$

In ALGPR, the procedure $isPrivate$ checks whether a suppression vector preserves δ-$privacy$ or not. More specifically, the process considers for all possible output state sequences based on the suppression vector and any sensitive context at any time of the corresponding layer, whether the system can preserve privacy or not of this layer.

The training process can be done on the trusted third-party server. After the training process, we can get several chains and suppression vectors of the context hierarchy figured in Figure 2, M_1, M_2, M_3, M_4 and $\vec{P_1}, \vec{P_2}, \vec{P_3}, \vec{P_4}$.

C. Checking

For the context hierarchy, FDH uses a hierarchical hybrid check to make final decision for each current context. The check starts from the lowest layer to the highest layer and each layer utilizes the hybrid check[6] to make interim decision. Specifically, if the context of the lowest layer cannot be released by the hybrid check, FDH then considers its ancestor contexts until it finds one which can be released. If there is no context can be release, FDH outputs the symbol $\perp$ for the current context. This recursive process can be done along the vertical direction as shown in Figure 2. The hybrid check for each layer first conducts two privacy checks which are probabilistic check and simulatable check, and then chooses the decision with the higher utility. The hybrid check preserves privacy against strong adversaries who know the system and the Markov chains of the corresponding hierarchy. In the following, we briefly introduce the probabilistic check and the simulatable check and describe how to choose the better one. Note that both checks are conducted for a specific layer of the hierarchy.

- *Probabilistic check*: The probabilistic check makes decisions according to the suppression vector of a specific layer of the context hierarchy. It flips a coin with a probability of the outcome heads specified by the suppression probability of the corresponding context.
- *Simulatable check*: The simulatable check makes decisions only based on the information available to the adversary at that time, namely, the Markov chains of the specific layer of the context hierarchy and the output contexts $o_{l,1}, ..., o_{l,t'-1}$. The current state is ignored. The simulatable check considers for any possible state $c_{l,j}$ at time t', whether releasing $c_{l,j}$ preserves δ-privacy or not. If some state can be released, the simulatable check decides to release the current context.

After the probabilistic check or simulatable check, the hybrid check computes the utility. The utility of the probabilistic check is based on the corresponding suppression vector and the utility of the simulatable check is computed according to the process given in[6]. Then the hybrid check chooses the one with the higher utility.

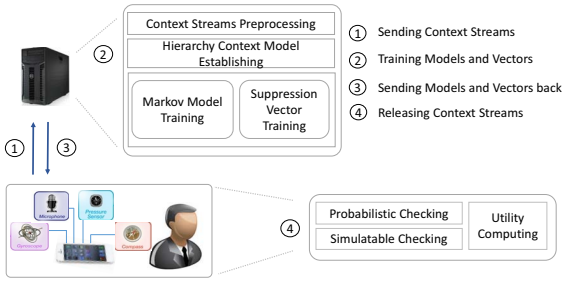

Figure 3. FDH in action

D. FDH in action

In order to employ FDH to help a user make online decisions, we divide the process of FDH into four parts.

- *Sensing context streams*: In order to train appropriate Markov chains and suppression vectors for each layer based on the context hierarchy, we need to collect sufficient and useful historical user contexts. These contexts are collected from mobile devices. The collection process must last for more than one month.
- *Training process*: The raw contexts are provided to the trusted third-party server. The server first preprocess data by removing noisy data, duplicate data and so on. Then the server uses the processed data to do the training work. Note that only sufficient historical contexts can model a user's behaviors and activities well. So the training process on the server is not necessary everyday. When the sensors have gathered sufficient historical contexts, the server executes this process once. The chain is trained based on the recent data, and the outdated data are ignored.

- *Returning the training results*: After the server finishes the training process, the Markov chains and suppression vectors for each layer of the context hierarchy have been produced. In order to utilize them to do privacy checks on mobile devices, the server returns them to the mobile devices. Because we train the chains and vectors for a single day, the size of results may not be very big and they not consume too much bandwidth to deliver across the network.
- *Online deciding*: The core of the online deciding part is the privacy check. The input of the check on mobile devices are the Markov chains, suppression vectors, the set of sensitive contexts and the privacy parameter. The set of sensitive context and the privacy parameter are configured by users. Then the hierarchical hybrid check starts from the lowest layer of the context hierarchy to the highest layer. For each layer, two kinds of privacy checks introduced above separately decides whether to release the corresponding context of the current context or suppress it and computes the utility. In the end, the hybrid check chooses the output with higher utility as the interim decision. If an interim decision is release or each layer is considered, the check process for the current context is finished.

Note that FDH considers the Markov chains and suppression vectors trained over a day, so it outputs context sequence for a single day. To get longer context sequences, one can just runs the system everyday. Because we assume there are not correlations across different days, so FDH can preserve δ-$privacy$ for a single day.

IV. Experimental Evaluation

In this section, we evaluate the performance of FDH in terms of time consumption of training and the privacy check, memory consumption of the privacy check and the number of contexts released as ancestor contexts of a user. In the following, we first introduce the experimental design and then discuss the experimental results.

A. Experimental Design

In our experiments, we evaluate FDH using the subset of the data collected for Reality Mining study. The subjects from this study consist of 106 students and staff at MIT. The data is recorded by Nokia 6600 smartphones during the months between September 2004 and June 2005[18]. The passive behavior logs contain locations, proximity to others, activities, transportation mode, etc. We use the location traces of 99 users who have at least 1 month of data because location represents the most complete and fine-grained context in the data set. Each location attribute consists of cell number and tower number. There is no overlap between the tower numbers of two locations with different cell numbers.

For each user, we construct a context hierarchy and train Markov chains on the first half of his trace; the remaining half is used for evaluation. Each location hierarchy contains three layers: location states with tower numbers in the lowest layer, location states with cell numbers in the middle layer, and the location representing all the locations in the top layer. Based on the context hierarchy of a user, we can train three different Markov chains and suppression vectors.

Unless stated otherwise, we choose $\delta = 0.25$, granularity of $d = 10$, and choose the home location of a user as the sensitive context.

We measure the ratio of the number of release states which is a weighted sum and all the contexts in the second half of the trace. The weight of a context is assigned according to the layer it belongs to. The weight for the bottom layer is 1, and it decreases by 0.5 while the layer goes up one.

We run our experiments on Intel Core 2.4GHZ Processor. To measure the overhead of FDH on smart phones, we also conduct experiments on a Google Nexus 5 phone.

B. Evaluation Results

1) Performance of FDH: First, we focus on the efficiency of the training process, including training Markov chains and training suppression vectors, with or without the context hierarchy (that is, just the lowest layer in the context hierarchy). We also measure the time consumption of online context release process using hybrid check with or without the context hierarchy. Table I shows the average time of training and release processes of all users.

Table I
Average processing times

Method	Training	Release	
	PC	PC	Phone
No Hierarchical Check	8min	$\leq$30ms	$\leq$120ms
Hierarchical Check	12min	$\leq$45ms	$\leq$210ms

The training process is finished within 12 min. In comparison, the time consumption of training with the context hierarchy of three layers is 50% higher than that without the context hierarchy.

The time consumption of release process is very little both on a PC and on a mobile phone. For example, the hierarchical check takes less than 45 ms on a PC and less than 210 ms on a mobile phone. Overall, the performance of FDH shows it is practical on a smart phone.

We find in this experiment, during the full process of FDH, the release part imposes negligible time consumption. Thus, it is critical to design a better algorithm to reduce the overall time consumption.

2) Sensitive contexts released: In[6], each context is either released or suppressed. However, in FDH, a user has more choices because of the context hierarchy. If the context cannot be released, the user can consider its ancestor contexts in the context hierarchy. Note that the checks in[6] are equivalent to those in FDH on the lowest layer of the context hierarchy. We count the number of contexts

which can be released as their ancestor contexts but can not be released as itself originally. We conduct two sets of experiments: in the first set, we choose three sensitive contexts for a user at random; in the second set, we choose five sensitive contexts for a user at random. Note that there are more than three or five sensitive states in each trace since states specify context and time.

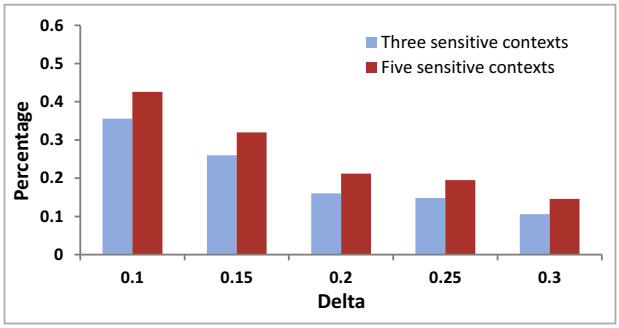

Figure 4. Comparison between different number of sensitive contexts

As shown in Figure 4, for both experiments, FDH can release a few of contexts in the form of their ancestor contexts. In average, FDH releases 42.6% of originally suppressed contexts at most and 10.6% at least. Given a fixed number of sensitive contexts, the percentage of the contexts released as their ancestors decreases while the value of δ increases. This is mainly because with weaker privacy guarantee, more contexts can be released as themselves. In addition, the percentage for the case of five sensitive contexts is more than that for the case of three sensitive contexts. That is to say, FDH is more suitable when there are more sensitive contexts.

3) Memory consumption: We then evaluate the memory consumption of FDH on a mobile phone. We conduct two sets of experiments: in the first set, we choose three sensitive contexts for a user at random; in the second set, we choose five sensitive contexts for a user at random. For each set, we change the values of δ.

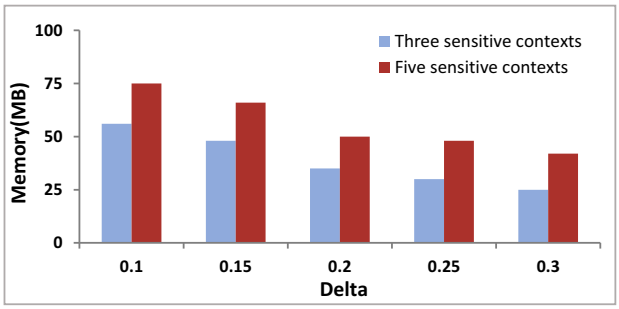

Figure 5. Comparison between different number of sensitive contexts

As shown in Figure 5, the memory consumption of check is at most 75MB, occupying less than 0.1% of the whole memory. The memory consumption for more sensitive contexts is more than that for fewer sensitive contexts. This is mainly because more sensitive contexts need more computation for each layer of the hierarchy. We also find that the memory consumption for bigger δ is more than that for smaller δ. The reason is that the hybrid check may consider more layers over the hierarchy for bigger δ.

4) Privacy-utility tradeoff: We evaluate the tradeoff between the privacy level and utility by varying the values of δ. We conduct two experiments: one for the hierarchical check on the hierarchy context model and another for privacy check only on the lowest layer. In both experiments, we choose a user's home location to be the sensitive context. Figure 6 shows the utilities for different values of δ.

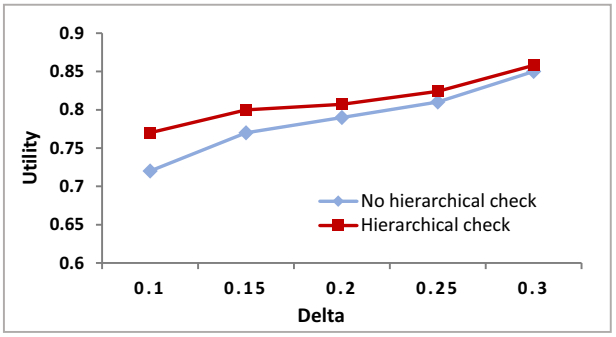

Figure 6. Privacy-utility tradeoff

We find that the utility increases with the increasement of δ. Since the privacy level decreases with the increasement of δ, there is a tradeoff between the privacy level and utility. On the other hand, we find the utility of hierarchical check based on hierarchy context model is higher than that only on the lowest layer. We also note that for both experiments, the overall decrease in utility is small. This implies that we can provide strong privacy guarantees (by choosing a smaller value of δ) without sacrificing too much utility.

V. Related Work

There are a lot of prior work on releasing context streams while preserving privacy. Anonymity-based technologies are widely used. k-anonymity[19] produces information for each person contained in the release that can not be distinguished from at least k-1 individuals whose information also appear in the release. The ℓ-diversity model[20] is an extension of the k-anonymity model which reduces the granularity of data representation. However, these anonymity-based technologies are susceptible to many attacks, especially when background knowledge is available to an attacker.

There have been some work to protect privacy against adversaries knowing background knowledge of users [6], [21]. Götz et al.[6] consider the adversary which is strong enough to know the system and the temporal correlations

in the form of a Markov chain. They provide δ-$privacy$ to explore a good tradeoff between privacy and utility. It supports online deciding whether to release or suppress the most fine-grained context by exploiting temporal correlations among contexts. Though δ-privacy achieves efficient control of sensitive contexts, it does not consider the intrinsic and naturally-available hierarchical relation among contexts.

FDH integrates the hierarchy context model into δ-privacy. It provides more fine-grained control over the context of different granularity instead of only releasing or suppressing the context itself. FDH can improve the quality of context-aware applications by releasing its ancestor when a context is suppressed in δ-privacy.

VI. Conclusion and future work

We focus on the problem of releasing user context streams while preserving privacy. FDH provides hierarchical hybrid privacy check to decide for each current context which of its ancestors in the context hierarchy to release or suppress it if no such ancestor exists. Our experiment evaluation on real context traces demonstrates that FDH can release a few suppression contexts in δ-privacy with limited cost and obtain relatively high utility while providing fine-grained privacy control for hierarchical contexts.

In FDH, we use the Markov chain (the probability of the next state depends only on the current state) to describe the frequencies of contexts and temporal correlations. However, more higher order Markov models may be more appropriate[22]. We will expand FDH by utilizing higher order Markov models.

In FDH, one context may be sensitive and considered as privacy. However, sometimes one context is not sensitive, but a behavior pattern including the context may be sensitive. For example, sending messages at 9:05 am is not sensitive, but the behavior pattern consisting of attending a meeting at 9:00 am, sending messages at 9:05 am and attending a meeting at 9:06 am is sensitive to a user. In the future, we will consider such privacy and use technologies like probabilistic model checking to solve it.

Acknowledgment

This work is supported by the National 973 Program of China (2015CB352202) and the National Science Foundation of China (61272047, 91318301, 61321491).

References

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, 2010.

[2] M. F. Mokbel and J. J. Levandoski, "Toward context and preference-aware location-based services," in *Proceedings of the Eighth ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM, 2009, pp. 25–32.

[3] E. Valavanis, C. Ververidis, M. Vazirgianis, G. C. Polyzos, and K. Norvåg, "Mobishare: sharing context-dependent data & services from mobile sources," in *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*. IEEE, 2003, pp. 263–270.

[4] G. Chen, D. Kotz *et al.*, "A survey of context-aware mobile computing research," Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, Tech. Rep., 2000.

[5] J. Y. Tsai, P. G. Kelley, L. F. Cranor, and N. Sadeh, "Location-sharing technologies: Privacy risks and controls," *ISJLP*, vol. 6, p. 119, 2010.

[6] M. Götz, S. Nath, and J. Gehrke, "Maskit: privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 289–300.

[7] G. Biegel and V. Cahill, "A framework for developing mobile, context-aware applications," in *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pp. 361–365.

[8] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, 2010.

[9] A. Mannini and A. M. Sabatini, "Accelerometry-based classification of human activities using markov modeling," *Computational intelligence and neuroscience*, vol. 2011, p. 4, 2011.

[10] G. Bieber, J. Voskamp, and B. Urban, "Activity recognition for everyday life on mobile phones," in *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, ser. Lecture Notes in Computer Science, C. Stephanidis, Ed.

[11] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 5–20.

[12] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, pp. 1554–1563, 1966.

[13] L. E. Baum, J. A. Eagon *et al.*, "An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology," *Bull. Amer. Math. Soc*, vol. 73, no. 3, pp. 360–363, 1967.

[14] L. E. Baum and G. R. Sell, "Growth transformations for functions on manifolds." *Pacific J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[15] L. R. Welch, "Hidden markov models and the baum-welch algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, pp. 10–13, 2003.

[16] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[17] A. Arasu, M. Götz, and R. Kaushik, "On active learning of record matching packages," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 783–794.

[18] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[19] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[20] C. C. Aggarwal and S. Y. Philip, *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[21] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Privacy Enhancing Technologies*. Springer, 2006, pp. 393–412.

[22] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlos, "Are web users really markovian?" in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 609–618.