POSTER: Model-Based Context Privacy for Personal Data Streams

Supriyo Chakraborty,
Kasturi Rangan Raghavan,
Mani Srivastava
UCLA
{supriyo, kasturir, mbs}@ucla.edu

Harris Teague
Qualcomm Inc.
hteague@qualcomm.com

ABSTRACT

Smart phones with increased computation and sensing capabilities have enabled the growth of a new generation of applications which are organic and designed to react depending on the user contexts. These contexts typically define the personal, social, work and urban spaces of an individual and are derived from the underlying sensor measurements. The shared context streams therefore embed in them information, which when stitched together can reveal behavioral patterns and possible sensitive inferences, raising serious privacy concerns. In this paper, we propose a model based technique to capture the relationship between these contexts, and better understand the privacy implications of sharing them. We further demonstrate that by using a generative model of the context streams we can simultaneously meet the utility objectives of the context-aware applications while maintaining individual privacy. We present our current implementation which uses offline model learning with online inferencing performed on the smart phone. Preliminary results are presented to provide proof-of-concept of our proposed technique.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Information flow controls

General Terms

Security

Keywords

Privacy, Context Streams, Dynamic Bayesian Network, Context-Awareness, Information leakage

1. INTRODUCTION

Smart phones are increasingly being used to sense our personal, social, work and urban spaces. There exists a rich body of prior work on the design of context-sensing algorithms and applications that leverage the multitude of sensors available on these phones. These algorithms use sensor measurements to infer a variety of contextual information like activity states (e.g., walking, running, sitting) [3, 10], semantic locations (e.g., work, home, office) [7], social neighborhoods (e.g., with friends, in a meeting) [9] etc. Recent research in this direction has been to incorporate the context inferencing engine as a service within the operating system itself [4, 8]. The availability of a rich set of user contexts has led to the design of applications which continuously use the information to provide context-aware personalization. However, the shared context streams are extremely personal and in addition to the desired utility for the application can reveal sensitive behavioral information about the user.

In this paper, we consider the problem of sharing personal context streams with untrusted applications such that on one hand, the desired *utility* in the form of application personalization is achieved, while on the other hand, individual *privacy* is maintained by preventing the release of sensitive contexts. To this end, we propose a model-based risk analysis framework that enables the user to track an adversary's (in the case the untrusted application) *information gain* over the sensitive contexts as other contexts are shared over time. An adversary's information gain is the decrease in uncertainty about the user being in a sensitive context based on the observations and indicates the risk of sharing. Based on the risk evaluation, the user can decide to either release or suppress the current context.

We build on top of prior work which aims to model the temporal correlation between the various contexts using a Markov chain [6]. However, such a model does not allow one to capture the relation between the different contexts, that exists, independent of their temporal dependence. We propose to use a more general graphical model in the form of a Dynamic Bayesian Network (DBN) which allows us to capture both the joint distribution between the different contexts and also their evolution over time. Our framework comprises of a context engine which runs continuously on the phone, monitors sensor data and provides inferred contexts. These contexts are then evaluated using the DBN model and current adversarial knowledge to quantify the amount of information they reveal about any sensitive context. If the information gain of the adversary, due to the release, is within user-specified tolerable limits the context is released else it is suppressed.

The rest of the paper is organized as follows. In Section 2 we define the system model as well as the adversary model. This is followed by Section 3 where we define our privacy and utility objectives. We then present our implementation and preliminary results in Sections 4 and 5 respectively. We conclude in Section 6.

CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.
ACM 978-1-4503-1651-4/12/10.

2. SYSTEM MODEL

We closely follow the system model in [6] which resembles today's sensor equipped smart phones running context-aware applications. Untrusted applications, requesting data, have access only to the higher level user contexts, provided by our model-based framework, and not to raw sensor data. A context engine runs on the phone and provides contexts $c_1, c_2, \ldots$ at discrete time instants. We denote by C, the set of all possible contexts a user can be in. Upon observing $c_t \in C$ at time t, we produce an output o_t which is either equal to the true observation c_t, or to the synthetic observation "close" to the true value c_t' or completely suppressed, denoted by $\perp$. We denote by $o_{1:t}$ the vector of all outputs $o_1, o_2, \ldots, o_t$ up to time t. In this work, we do not consider synthesis as a possible mechanism for generating value for o_t. However, the use of a generative model such as the DBN allows us to synthesize contexts by sampling their joint distribution. In many cases, the synthetic context released can continue to provide applications with the required utility while still maintaining user privacy. We defer this extension to future work.

A user specifies sensitive contexts, which we denote by the set $C_s \subseteq C$. The value of o_t depends on the DBN, the released outputs $o_{1:t-1}$, and the user specified sensitive contexts C_s. The elements of C form the nodes of the DBN. While [6] deals with only the temporal correlation between the contexts, DBN is a more general model which captures the joint distribution between contexts at a particular time snapshot as well as their correlation over time. Finally, the order of the DBN also allows us to model the temporal relevance of contexts and age it over time. The process of aging context sensitivity allows us to re-initialize the system and share contexts again after a particular time period has elapsed. This prevents the model from being conservative after a few releases thus maintaining its utility over time. Let $X_1, X_2, \ldots, X_t$, denote random variables produced by sampling the DBN, where $X_i \in C \cup \{\perp\}$. We want to evaluate $Pr(X_{t+k}|o_{1:t})$ where $o_{1:t}$ is a vector containing all the system outputs till time t. The index $k > 0$ allows us to make predictions in the future, whereas a negative $0 \leq k \leq t$ allows us to compute posterior over a past state.

Adversary Model

We assume a strong Bayesian adversary who has complete knowledge of the user DBN and also has access to the output vector $o_{1:t}$. Based on the DBN, the adversary maintains a prior $Pr(X_t = c_i)$ and upon observing the output they infer as much as possible about the sensitive states and update the posterior belief as $Pr(X_t = c_i|o_{1:t})$. To beat the adversary the goal is to track his information gain over time by continual update of the DBN based on the output values.

3. PRIVACY AND UTILITY

Based on the adversarial model, we define a notion of $\delta-$privacy similar to [6]. It is a simple policy which states that for all possible outputs $o_{1:t}$, for all times t, and over all possible user specified sensitive states $s \in C_s$, release information only if

$$Pr(X_t = s|o_{1:t}) - Pr(X_t = s) \leq \delta. \qquad (1)$$

This forms our privacy objective. Intuitively, the utility objective is to maximize the release of the true states, while meeting the objectives in Eqn. 1. Again, the generative nature of the DBN allows us to find contexts which are "close" to the current context and does not reveal information about the sensitive contexts. These synthesized contexts could instead be released to maximize the utility objective.

Table 1: DBN for a single user with 10-fold cross validation.

Parameter	Value
Number of data points	3000
Algorithm Used	Bayesian Search
Time to learn	3.4 seconds

4. IMPLEMENTATION

We have a partial implementation of the proposed system on smart phones running Android OS 2.2 or higher.

DBN Learning: The DBN model learning is done offline on the cloud as it is computationally intensive to run it on the phone. We use the GeNIe+SMILE package [1] for learning the model which is then transferred to the phone in the Bayesian interchange file format (BIF). For probabilistic inferences on the phone, we port the Java Bayes package [2] to the Android platform. Specifically, we use the variable-elimination algorithm implemented in the package for computing the posterior probabilities.

Context Framework: We are currently in the process of creating a context inferencing framework on the mobile phone. However, in this work, for evaluating risk, we have used the reality mining dataset [5] for deriving our context stream. The dataset contains behavioral data logged continuously by a group of 94 subjects over a period of 9 months. Among other details it includes the distribution of user location (e.g. *home, work, elsewhere*), time of day, and phone activity in the form of applications launched (e.g. *phone, browser, camera*). We define the set of possible contexts $C =$ {*home, work, elsewhere, Browser, ClockApp, ContextLog, Camera, MediaGallery, Calendar, Phone*}. An applications context can be in either of two states $\{launched, not\_launched\}$. Similarly, except *home*, the other location contexts can be either *true* or *false*. Based on the dataset, we have an additional *unknown* value for location *home*.

Scenario: We consider a stream of time indexed tuples from a single user derived by pre-processing the actual dataset. Each tuple includes the following fields:

$$(\text{ home, work, elsewhere, app1, } \ldots, appN)$$

where $app1, \ldots, appN$ are the different application contexts. The offline training phase comprises of a 10-fold cross validation to learn the DBN. The parameters of the learning process for a specific network instance is shown in Table. 1. For space constraints we present the Bayesian network for a single time slot and not the unrolled DBN over multiple time slots. We assume that the user is willing to respond to queries about the applications he uses but would want to protect his location from being inferred when he is at *home*. Thus, the set $C_s = \{home\}$.

5. PRELIMINARY RESULTS

We learn a DBN as shown in Fig. 1. The user selects *home* as his sensitive location. The tolerance parameter $\delta = 0.36$.

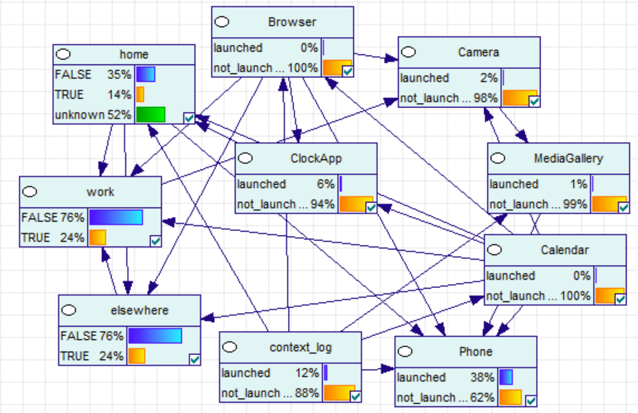

Figure 1: Learned DBN where *home* is the sensitive state. The prior probability $Pr(home = True) = 0.14$. The value of $\delta = 0.36$.

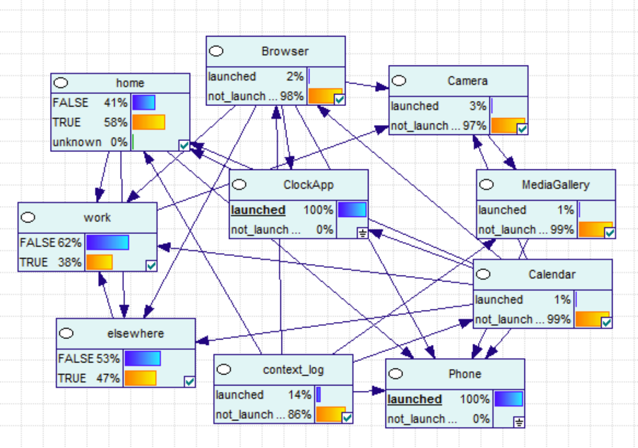

Figure 2: After assertion of *Phone* at $t = 1$, posterior changes to $Pr(home|Phone) = 0.26$. At $t = 2$, if we assert *ClockApp*, posterior changes to $Pr(home|Phone, ClockApp) = 0.58$.

Assuming a Bayesian adversary, the goal is to probabilistically suppress the release of an application name depending on computed posterior probability. Initially, the prior known to the adversary is $Pr(home = true) = 0.14$. At $t = 1$, the user launches the Phone application, accordingly the inference engine computes the posterior as $Pr(home = true|Phone = launched) = 0.26$. Since the difference is less than δ, we release the application name and update the posterior accordingly to reflect the adversarial knowledge. However, at $t = 2$, user launches the *ClockApp*, and the corresponding posterior is $Pr(home = true|Phone = launched, ClockApp = launched) = 0.58$. The difference between the posterior and prior is now greater than δ, and we therefore suppress the application name. Note, while we show our computation for a snapshot Bayesian network in Fig. 1, we can evaluate the same on a DBN.

6. CONCLUSION

We presented a model-based privacy risk analysis framework that includes at its core a Dynamic Bayesian Network to track an adversary's belief in the value of sensitive contexts as other context streams are shared and as time passes. We then presented a sharing policy that uses the computed beliefs to ensure privacy of sensitive contexts.

The framework presented makes an assumption that the underlying context stream derived from the sensor data has a predictable model, as demonstrated in Fig. 1. While DBNs are a well-known class of graphical models to succinctly represent joint distributions over variables, and relatively efficient algorithms for learning and inference on DBNs are known, it is possible that no tractable DBN representations exist that can capture the relationship between the application requested contexts and the sensitive contexts, for e.g. when the dependence manifests itself only after long period of time.

In the future, we would like to build on top of this proposed model. For usability reasons, it will be useful to provide users with guidance regarding the choice of the privacy parameter δ. Another extension, already discussed in the paper, will be to exploit the generative power of the DBN model to evaluate the efficiency of generating synthetic data as output, instead of suppression, as a privacy technique.

7. REFERENCES

[1] Genie and smile. `http://genie.sis.pitt.edu/`.

[2] Java bayes. `http://www.cs.cmu.edu/javabayes/`.

[3] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive'04*, pages 1–17, 2004.

[4] D. Chu, A. Kansal, J. Liu, and F. Zhao. Mobile apps: it's time to move up to condos. In *Proceedings of the 13th USENIX conference on Hot topics in operating systems*, HotOS'13, pages 16–16, 2011.

[5] N. Eagle, A. S. Pentland, and D. Lazer. Inferring Social Network Structure using Mobile Phone Data. In *Proceedings of the National Academy of Sciences (PNAS)*, pages 15274–15278, 2008.

[6] M. Götz, S. Nath, and J. Gehrke. Maskit: privately releasing user context streams for personalized mobile applications. In *SIGMOD*, pages 289–300, 2012.

[7] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, 2010.

[8] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, 2010.

[9] T. Nicolai, E. Yoneki, N. Behrens, and H. Kenn. Exploring social context with the wireless rope. OTM'06, pages 874–883, 2006.

[10] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, Mar. 2010.