# MaskIt: Privately Releasing User Context Streams for Personalized Mobile Applications

Michaela Götz[*]
Twitter Inc.
mila@twitter.com

Suman Nath
Microsoft Research
sumann@microsoft.com

Johannes Gehrke
Cornell University
johannes@cs.cornell.edu

## ABSTRACT

The rise of smartphones equipped with various sensors has enabled personalization of various applications based on user contexts extracted from sensor readings. At the same time it has raised serious concerns about the privacy of user contexts.

In this paper, we present MaskIt, a technique to filter a user context stream that provably preserves privacy. The filtered context stream can be released to applications or be used to answer queries from applications. Privacy is defined with respect to a set of sensitive contexts specified by the user. MaskIt limits what adversaries can learn from the filtered stream about the user being in a sensitive context – even if the adversaries are powerful and have knowledge about the filtering system and temporal correlations in the context stream.

At the heart of MaskIt is a privacy check deciding whether to release or suppress the current user context. We present two novel privacy checks and explain how to choose the check with the higher utility for a user. Our experiments on real smartphone context traces of 91 users demonstrate the utility of MaskIt.

## Categories and Subject Descriptors

H.3.5 [**INFORMATION STORAGE AND RETRIEVAL**]: Online Information Services—*data sharing*

## General Terms

Algorithms, Security

## 1. INTRODUCTION

Mobile devices today are increasingly equipped with a range of sensors such as GPS, microphone, accelerometer, light, and proximity sensors. These sensors can be effectively used to infer a user's context including his location (e.g. at home or in the office) from GPS, transportation mode (e.g. walking or driving) from accelerometer, social state (e.g. alone or in a group) from the microphone, and other activities (e.g. in a meeting) from a combination of sensors. Consequently, a large and increasing number of applications in popular smart phone platforms such as the iPhone, the Android, and the Windows Phone utilize user contexts in order to offer personalized services. Examples of such applications include $GeoReminder$ that notifies the user when he is at a particular location, $JogBuddy$ that monitors how much he jogs in a day, $PhoneWise$ that automatically mutes the phone during meetings, $SocialGroupon$ that delivers coupons or recommendations when he is in a group of friends, etc.

However, these context-aware mobile applications raise serious privacy concerns. Today, people already believe that risks of sharing location information outweigh the benefits in many location-based services [34]. One reason why risks are high is that many mobile applications today aggressively collect much more personal context data than what is needed to provide their functionalities [7] (for example, a calculator application might send the user's location to an advertisement server). Moreover, applications rarely provide privacy policies that clearly state how users' sensitive information will be used, and with what third-parties it will be shared. To avoid the risks, a user can decide not to install these application or not to release any context information to them (by explicitly turning off sensors); but then the user might not be able to enjoy the utility provided by these applications. In order to explore a better tradeoff between privacy and utility, we can let the user control at a fine granularity when and what context data is shared with which application [18, 34]. For example, a user might be okay to release when he is at lunch but he might be hesitant to release when he is at a hospital. With such fine-grained decisions, a user can choose a point in the privacy-utility tradeoff for an application and can still enjoy its full functionality when he chooses to release his context information or when his context information is not actually needed.

To support such fine-grained control, we need to answer the question: *When and what context should be suppressed* to preserve privacy? A naïve approach, that we call MaskSensitive, is to let the user specify sensitive contexts and to simply suppress those. This, however, does not necessarily prevent an adversary from *inferring* sensitive contexts. One reason why an adversary can infer suppressed sensitive contexts is that the suppression itself leaks information. Consider a user who suppresses his context if and only if he is playing a video game at work. An adversary knowing the suppression rule can infer exactly when he is playing a game at work. In general, we want to guard against leakage attacks from *adversaries knowing the suppression system.* Such adversaries are powerful and can reverse-engineer the system in order to infer information about suppressed contexts. Protecting against them follows Shannon's maxim "The enemy knows the system," and does not rely on privacy through obscurity.

Another way an adversary can infer a sensitive context is by ex-

[*] Work done while at Cornell University.

*SIGMOD '12,* May 20–24, 2012, Scottsdale, Arizona, USA.

ploiting temporal correlations between contexts. Consider a user who suppresses his location when he is at a hospital. This, however, might not be sufficient: when he releases his non-sensitive context while he is driving to the hospital, the adversary can infer where he is heading. Similarly, when he releases the use of a hospital finder app, the adversary can again infer where he is heading. In these cases the sensitive context can be inferred from its dependence on non-sensitive contexts. In general, we want to guard against inference attacks from *adversaries knowing temporal correlations*. Such adversaries are realistic because human activities and contexts exhibit daily and weekly patterns. For example, Bob is at work at 9am during the week and he always picks up his children from daycare after returning from work. Previous work has shown that human behavior and activities can be modeled well with a simple Markov chain [15, 24]. We use the same approach and model the user behavior as a Markov chain over contexts with transition probabilities that generates the stream of contexts. A Markov chain captures frequencies of contexts and temporal correlations. Adversaries can gain knowledge about patterns and frequencies of contexts from observing a person to create a rough daily schedule or by using common sense; for example, knowing that Bob works full time at a bakery, the adversary can guess that he is most likely to be at work at 9am. An adversary can also extract patterns from the sequence of contexts released to applications. We consider strong adversaries knowing the Markov chain of a user and weaker adversaries with only limited information about the Markov chain.

In the presence of such adversaries, the aforementioned question needs to be reformulated as *when and what context should be suppressed in order to protect privacy against adversaries knowing temporal correlations and also the system making this decision?* Here, privacy is defined with respect to a set of sensitive contexts. Users can decide on the sensitivity of contexts (for example, include “at the hospital” but not “walking the dog”) with the help of special tools (see for example [33]). Guaranteeing privacy means to limit what an adversary can learn about a user being in a sensitive context at some point in time from the released sequence of contexts. By looking at the released contexts and combining them with his background knowledge, an adversary should not be able to learn that a user was/is/will be in a sensitive state. Our experiments show that the MaskSensitive approach does not meet this requirement: more than half of the masked sensitive states constitute privacy breaches, i.e., upon observing the output generated by MaskSensitive, an adversary can use his background knowledge about a user’s Markov chain to gain much confidence in the fact that the user is in a sensitive state.

In this paper, we propose MASKIT, a system that addresses the above question with two novel privacy checks deciding in an online fashion whether to release or suppress the current state of the user. The *probabilistic check* flips for each context a coin to decide whether to release or suppress it. The bias of the coin is chosen suitably to guarantee privacy. The *simulatable check* makes the decision only based on the released contexts so far and completely ignores the current context. That way, the decision does not leak additional information to the adversary. Both checks provably provide privacy, but interestingly their relative benefit varies across users —there are situations where the probabilistic check provides higher utility than the simulatable check and vice versa. We explain how to select the better check among the two for a given user.

Both checks provide privacy against very strong adversaries who know the system and the Markov chain modeling a user and his frequent patterns. We also consider weaker adversaries with less background knowledge about the user model. Protecting against these adversaries is challenging because they can learn and gain additional knowledge either from other sources or from the released contexts.[1] We explain how to adapt our checks to preserve privacy against weaker adversaries as they learn.

There is large body of prior work on privacy-preserving publishing of location streams. Most work does not consider adversaries knowing temporal correlations. The only system that we are aware of that provably protects privacy in location streams against adversaries knowing the system and temporal correlation is limited to a *single* type of temporal correlation based on the maximum velocity of a user [9]. It is vulnerable to attacks from adversaries knowing frequencies (e.g., the user is never at home at 2pm) or other temporal correlations (e.g., the distribution of time between two consecutive user locations or the average velocity). Schemes based on cryptographic protocols [10, 28] provide a strong privacy guarantee, but they cannot release streams of user contexts. To the best of our knowledge, our work is the first to release user context streams while protecting privacy of sensitive contexts against powerful adversaries knowing the system and various temporal correlations including typical user behavior (the user gets up every day at 6am) and correlations (after going to the doctor the user is likely to go to the pharmacy). Moreover, previous work offering privacy guarantees for streams focuses exclusively on locations. Our work can handle more general contexts including the social state, the transportation mode, and activities as we illustrate in our experiments.

We have evaluated MASKIT on a PC as well as on a smart phone, with real public traces from 91 human subjects over the course of nine months, representing user contexts over 266,000 hours. Our evaluation shows that we do not have to pay a high price in terms of utility and efficiency for the privacy guarantee: MASKIT releases nearly as many states as the MaskSensitive approach (and only 7% fewer states in the worst case), and MaskSensitive does not guarantee privacy. Moreover, the suppression decision incurs negligible overhead ($\leq$ 128ms on average on a smart phone) compared to the context extraction time of typically a few to tens of seconds.

In summary, the paper makes the following contributions.

- We present two privacy checks for deciding whether to release or suppress a user’s current context. They provably preserve the privacy of sensitive contexts against powerful adversaries knowing the system and the Markov chain of the user (Sections 3 and 4).
- We examine how the relative benefit of the two privacy checks varies across users, and we provide a superior hybrid privacy check (Section 5).
- We explain how to adapt these checks to protect privacy against adversaries who have limited knowledge about the user’s Markov chain but can potentially learn more about it over time (Section 6).
- We compare the privacy checks experimentally on real user context traces (Section 7).

We start by laying out the problem of privately releasing user context streams and describing the overall architecture of MASKIT.

# 2. PROBLEM STATEMENT

## 2.1 MASKIT

**System Model.** We assume a system that models today’s sensor-equipped smart phones running context-aware applications (e.g., those mentioned in Section 1). Untrusted applications access user contexts through MASKIT and do not have access to raw sensor

[1] This observation lead to attacks in micro-data publishing [19, 36].

data.[2] For energy-efficient support of continuously running applications, MASKIT senses user contexts $x_1, x_2, \ldots$ periodically at discrete points in time (like [2, 26]). Upon extracting a context $x_t$ at time $t$, MASKIT produces a privacy-preserving output $o_t$. Continuously running applications can subscribe to the full privacy-preserving context stream $o_1, o_2, \ldots$ MASKIT can also serve applications issuing sporadic queries over the stream (e.g. asking for the current context), although these applications are not the main focus of this work.

To compute $o_t$, MASKIT employs a check deciding whether to release or suppress the current context $x_t$. The check follows the "release or suppress" paradigm and restricts the output $o_t$ to be either the true state $x_t$ or the suppression symbol $\perp$, i.e., $o_t \in \{x_t, \perp\}$. [3] We make this restriction because it reflects standard access control mechanisms in existing phones and the modus operandi of many location-based mobile applications [34]. This restriction makes it easy to port existing applications to MASKIT—all that is necessary is the ability to deal with suppressed states in the stream.

**User Model.** We assume that a user's various contexts and transitions between them can be captured by a Markov chain $M$; i.e., the user behaves like a sample from $M$. Previous work has shown that human behavior and activities extracted from smartphone sensors can be modeled well with a simple Markov chain [15, 24]. Markov chains have also been used to model user behavior in other domains including computer-aided manufacturing [20], Web search [3, 16] and search in entity relation graphs [4]. The states in $M$ are labeled with contexts $\{c_1, \ldots, c_n\}$. The transition probability $a^t_{i,j}$ denotes the probability of the user being in context $c_j$ at time $t$ given that he is in context $c_i$ at time $t-1$. We use the term *state* to denote a user's context at a given time (e.g., at home at 9pm).

We consider a model over a day: the states in $M$ represent all possible contexts of a user in a day.[4] Each day, the user starts at the "start" state in $M$ and ends $T$ steps later in the "end" state. Here, $T$ denotes the sensing frequency. We denote by $X_1, \ldots, X_T$ random variables generated from $M$, each taking on the value of some context $c_i$. The independence property of Markov chains states that

$$\Pr[X_t = c_i | X_1, \ldots, X_{t-1}] = \Pr[X_t = c_i | X_{t-1}]. \qquad (1)$$

**Adversary Model.** We consider two types of adversaries. *Strong adversaries* know the Markov chain $M$ of a user. *Weak adversaries* initially have less knowledge about $M$; but they can learn more about $M$ over time. We further assume that adversaries can access the full output sequence generated by a general suppression system $\mathcal{A}$, and we assume the adversaries also know $\mathcal{A}$. [5] In the following, We assume that the adversaries apply Bayesian reasoning. Based on the Markov chain, adversaries have a prior belief about the user being in context $c_i$ at time $t$, denoted by $\Pr[X_t = c_i]$, where the randomness comes from the process $M$ generating $X_1, \ldots, X_T$. Upon observing a released output sequence, they infer as much as possible about contexts and update their belief. The posterior belief, denoted by $\Pr[X_t = c_i | \mathcal{A}(\vec{x}) = \vec{o}]$ is computed by conditioning the prior belief on the observed sequence $\vec{o}$ that was generated from the user's sequence $\vec{x}$ by the system $\mathcal{A}$. The randomness comes from $M$ and $\mathcal{A}$. When it is clear which system $\mathcal{A}$ we are referencing, we drop it from our notation. More details about the computation of beliefs can be found the next subsection.

**Privacy.** Consider a user $u$ with a Markov chain $M$ over contexts $c_1, \ldots, c_n$. The user declares a subset of these contexts $S \subset \{c_1, \ldots, c_n\}$ as sensitive (e.g., by using special tools [33]). Informally, a released sequence $\vec{o}$ preserves privacy if the adversary cannot not learn much about the user being in a sensitive state from $\vec{o}$. That is for all sensitive contexts and all times we require the posterior belief about the user being in the sensitive context at that time not to be too much larger than the prior belief.[6]

DEFINITION 1. *We say that a system $\mathcal{A}$ preserves $\delta$-privacy against an adversary if for all possible inputs $\vec{x}$ sampled from the Markov chain $M$ with non-zero probability (i.e.,* $\Pr[\vec{x}] > 0$*), for all possible outputs $\vec{o}$* ($\Pr[\mathcal{A}(\vec{x}) = \vec{o}] > 0$)*, for all times $t$ and all sensitive contexts $s \in S$*

$$\Pr[X_t = s | \vec{o}] - \Pr[X_t = s] \leq \delta.$$

Note, that our privacy definition also limits what an adversary can learn about the user being in *some* (as opposed to a specific) sensitive context at a certain time. In general, for any subset $S' \subset S$, any $t$, any $\vec{o}$ preserving $\delta/|S'|$-privacy according to the above definition we have that $\Pr[X_t \in S' | \vec{o}] - \Pr[X_t \in S'] \leq \delta$. This is because $\Pr[X_t \in S' | \vec{o}]$ is equal to $\sum_{s \in S'} \Pr[X_t = s | \vec{o}]$. By the $\delta/|S'|$-privacy guarantee this is at most $\sum_{s \in S'} \delta/|S'| + \Pr[X_t = s]$ which is equal to $\Pr[X_t \in S'] + \delta$.

Furthermore, our privacy definition limits what an adversary can learn about the user being in a sensitive context in a time window. For a time window of length $\Delta_t$ any $\vec{o}$ preserving $\delta/\Delta_t$-privacy we have that the posterior belief formed after observing $\vec{o}$ of the user being in a sensitive context $s$ at *some* point in the time window is at most $\delta$ larger than his prior belief.

**Utility Goal.** We want to release as many states as possible, while satisfying the privacy goal. We measure the utility of a system for a user with chain $M$ as the expected number of released states in an output sequence. The randomness comes from $M$ and the system.

**The MASKIT System.** Algorithm 1 shows the MASKIT system. It takes as input the user's model $M$, sensitive contexts $S$, and the privacy parameter $\delta$. MASKIT learns $M$ from historical observations $x_1, x_2, \ldots$. The other two parameters can be configured by the user to obtain the desired level of privacy. At its heart, there is a privacy check deciding whether to release or suppress the current state. This privacy check supports two methods initialize and okayToRelease. After the initialization, MASKIT filters a stream of user contexts by checking for each context whether it is okay to be released or needs to be suppressed. MASKIT releases an output sequence "start", $o_1, \ldots, o_T$, "end" for a single day. We can use MASKIT repeatedly to publish longer context streams. It suffices to prove privacy of a single day due to our assumption that there are no correlations across days. Before describing the privacy check for a day, we lay the foundation for the privacy analyses and review Markov chains.

## 2.2 Preliminaries

This section forms the background of the adversarial reasoning. We roughly follow the notation of Manning and Schütze [23].

**Markov chains.** Markov chains constitute the background knowledge of our adversaries. Consider a Markov chain with random variables $X_1, \ldots, X_T$ each taking on a value in the set of contexts $C = \{c_1, \ldots, c_n\}$. A chain is not necessarily time-homogenous, i.e., the transition probability from one context to another may depend on the time. Thus we can view such a chain is as a DAG

[2] Trusted applications, however, can access raw data and contexts directly if needed.

[3] Other ways of sanitizing the output, e.g., generalization, are left for future work.

[4] To capture correlations across days, we can consider a larger model capturing a week or a month.

[5] This type of knowledge is often overlooked [35, 37].

[6] If the sensitivity of a context depends on the time then we can generalize this definition to sensitive *states*. Extending our system is straight-forward.

**Algorithm 1** System to generate $\delta$-private streams.

```
procedure MASKIT($\delta$, Markov chain $M$, sensitive contexts $S$)
    initialize($\delta, M, S$)
    $\vec{o}$ = "start"
    for current time $t \in [1, 2, \ldots, T]$ do
        $c_i$ = GETUSERCURRENTCONTEXT()
        if okayToRelease($c_i, t, \vec{o}$) then
            $o_t = c_i$
        else $o_t = \perp$
        $\vec{o} \leftarrow \vec{o}, o_t$
        Release $o_t$
    $\vec{o} \leftarrow \vec{o}$, "end"
```

with $T + 2$ levels, in which a state at level $t$ has outgoing edges to all states in level $t + 1$ (possibly with probability zero). At level 0 we have the "start" state and at level $T + 1$ we have the "end" state; Figure 1 shows an example. Note that states at different levels might carry the same context label. Thus, we can describe the Markovian process with transition matrices $A^{(1)}, \ldots, A^{(T+1)}$:

$$a^{(t)}_{i,j} = \Pr[X_t = c_j | X_{t-1} = c_i]$$

PROPOSITION 1. *The prior belief of an adversary (who knows a user's chain $M$) about the user being in a sensitive context $s$ at time $t$ is equal to*

$$\Pr[X_t = s] = (A^{(1)} \cdot A^{(2)} \cdots A^{(t)})_s.$$

The joint probability of a sequence of states is:

$$\Pr[X_1, \ldots, X_T] = \prod_{t=1}^{T} a^{(t)}_{X_{t-1}, X_t}$$

In general, we can compute the probability of transitioning from state $c_i$ at time $t_1$ to state $c_j$ at time $t_2$ efficiently:

$$\Pr[X_{t_2} = c_j | X_{t_1} = c_i] = (e_i A^{(t_1+1)} \cdots A^{(t_2)})_j \qquad (2)$$

where $e_i$ is the unit vector that is 1 at position $i$ and 0 otherwise.

**Hidden Markov Models.** Hidden Markov models help us understand how adversaries make inference about suppressed states. Each state has a distribution over possible outputs from a set $K = \{k_1, \ldots, k_m\}$. The output at time $t$ is a random variable $O_t$. The random variable $O_t$ is conditionally independent of other random variables given $X_t$. We define emission matrices $B^{(t)}$ as:

$$b^{(t)}_{i,k} = \Pr[O_t = k | X_t = c_i]$$

For a given output sequence $\vec{o} = o_1, \ldots, o_T$, we compute the conditional probability that at time $t$ the hidden state was $c_i$:

$$\Pr[X_t = c_i | \vec{o}] = \frac{\Pr[X_t = c_i, o_1, \ldots, o_{t-1}] \Pr[o_t, \ldots, o_T | X_t = c_i]}{\Pr[\vec{o}]}$$

We use the forward procedure $\alpha$ and the backward procedure $\beta$ to compute this ratio efficiently.

$$\alpha_i(t) = \Pr[X_t = c_i, o_1, \ldots, o_{t-1}] = \sum_j \alpha_j(t-1) a^{(t)}_{j,i} b^{(t-1)}_{j, o_{t-1}}$$

We initialize $\alpha_j(1) = a^{(1)}_{\text{"start"}, j}$ for all $j$.

$$\beta_i(t) = \Pr[o_t, \ldots, o_T | X_t = c_i] = \sum_j b^{(t)}_{i, o_t} a^{(t+1)}_{i,j} \beta_j(t+1)$$

We initialize $\beta_i(T + 1) = 1$ for all $i$. Putting everything together results in the following formula:

$$\Pr[X_t = c_i | \vec{o}] = \frac{\alpha_i(t) \beta_i(t)}{\sum_j \alpha_j(t) \beta_j(t)} \qquad (3)$$

**Algorithm 2** Probabilistic Privacy Check.

```
procedure initialize(($\delta, M, S$))
    $\vec{p} \leftarrow \arg\max_{\vec{p}}$ utility($\vec{p}$)
    subject to ISPRIVATE($\delta, \vec{p}, S, M$) = true

procedure okayToRelease($c_i, t', \cdot$)
    with probability $p_i^{t'}$ return false
    return true

procedure ISPRIVATE($\delta, \vec{p}, S, M$)
    for each $s \in S$ do
        for $t \in [T]$ do
            Compute prior $\Pr[X_t = s]$.
            for output sequences $\vec{o}$ do
                if $\Pr[\vec{o}] == 0$ then continue
                Compute posterior $\Pr[X_t = s | \vec{o}]$.
                if posterior − prior > $\delta$ then
                    return false
    return true
```

# 3. PROBABILISTIC PRIVACY CHECK

In this section we develop a probabilistic privacy check that specifies for each state $c_i$ at time $t'$ a suppression probability $p_i^{t'}$ with which this state is suppressed. With probability $1 - p_i^{t'}$, $c_i$ is released at time $t'$. Among all vectors of suppression probabilities $\vec{p}$ that preserve $\delta$-privacy, we seek one with the maximum utility. We measure utility as the expected number of released contexts:

$$\mathsf{utility}(\vec{p}) = \sum_{\vec{o}} \Pr[\vec{o}] |\{i | o_i \neq \perp\}| = \sum_{t' \in [T], i \in [n]} \Pr[X_{t'} = c_i](1 - p_i^{t'})$$

EXAMPLE 1. *Consider the Markov chain in Figure 2(a). Two states s, x are reachable from the "start" state with equal probability of* $1/2$. *Both immediately transition to the "end" state. The sensitive context is s. To achieve* $\delta = 1/4$*-privacy it suffices for the probabilistic check to suppress s with probability 1 and x with probability* $1/3$*: The prior belief of* $X_1 = s$ *is* $1/2$. *The posterior belief upon observing* $\perp$ *is* $\Pr[X_1 = s] p_s^1 / (\Pr[X_1 = s] p_s^1 + \Pr[X_1 = x] p_x^1) = 3/4$. *Suppressing s with probability* $< 1$ *breaches privacy. If s was ever released then the posterior belief would be 1 which is more than* $\delta$ *larger than the prior belief. Also, if x was suppressed with probability* $< 1/3$ *then the posterior belief of* $X_1 = s$ *upon observing* $\perp$ *would be more than* $\delta$ *larger than the prior belief. Thus,* $p_s^1 = 1, p_x^1 = 1/3$ *preserves privacy and maximizes utility.*

Usually, we expect to always suppress a sensitive state $s$ (unless it has a really high prior belief $\geq 1 - \delta$) and other states with sufficiently high probability so that upon observing $\perp$, an adversary is uncertain whether the suppressed state is $s$.

The probabilistic privacy check is outlined in Algorithm 2, where initialize formalizes the optimization problem of finding a suitable suppression probability vector $\vec{p}$. The okayToRelease method simply uses this vector to release or suppress current states. These two methods are used by the MASKIT system described in Section 2.

In the remainder of the section, we focus on the optimization problem in the initialize method. It makes use of the ISPRIVATE method that checks if a suppression vector $\vec{p}$ preserves $\delta$-privacy.

## 3.1 Checking a Suppression Vector

Following Definition 1, we compute whether a vector of suppression probabilities $\vec{p}$ preserves $\delta$-privacy as follows: We enumerate all possible output sequences $\vec{o}$ up to length $T$ and iterate over all times $t$ and all sensitive contexts $s$ to make sure that the posterior belief is at most $\delta$ larger than the prior belief. The process is shown in the Procedure ISPRIVATE in Algorithm 2.

**Details on computing beliefs.** The user's chain $M$ together with the probabilistic check using $\vec{p}$ form a hidden Markov model generating $\vec{x}$ as hidden states and $\vec{o}$ as output states. The hidden Markov model extends $M$ with emission matrices:

$$b_{i,k}^{(t)} = \Pr[O_t = k | X_t = c_i] = \begin{cases} p_i^t & \text{if } k = \perp \\ 1 - p_i^t & \text{if } k = i \\ 0 & \text{o.w.} \end{cases}$$

PROPOSITION 2. *An adversary who knows $M$ and the probabilistic check with suppression probabilities $\vec{p}$ computes his posterior belief simply as $\Pr[X_t = s|\vec{o}]$ in this hidden Markov model defined by $M$ and the emission matrices:*

$$b_{i,k}^{(t)} = \Pr[O_t = k | X_t = c_i] = \begin{cases} p_i^t & \textit{if } k = \perp \\ 1 - p_i^t & \textit{if } k = i \\ 0 & \textit{o.w.} \end{cases}$$

We can efficiently compute this posterior belief using Eq. (3).

## 3.2 Search Algorithm

We can solve the optimization problem of choosing the best suppression probabilities by iterating over all vectors $\vec{p}$ and checking if IsPrivate$(\delta, \vec{p}, S, M)$ returns true. For those passing the check we can compute their $\mathsf{utility}(\vec{p})$ and return the one with the maximum utility. This approach, however, is impractical: There is an infinite number of suppression probabilities and even if we discretized the space $[0, \ldots, 1] \to \{0, 1/d, 2/d, \ldots, 1\}$ there are still $d^{n \cdot T}$ many vectors to check. One might hope to apply efficient techniques for convex optimization. However, IsPrivate is neither convex nor concave. Thus, we cannot simply apply techniques for convex optimization. However, we can dramatically reduce the search space by exploiting the *monotonicity* property of privacy. To define monotonicity, we introduce a total ordering of suppression probabilities.

DEFINITION 2. *We say vector $\vec{q}$ that dominates $\vec{p}$, denoted by $\vec{p} \preceq \vec{q}$, if for all $i, t : p_i^t \leq q_i^t$.*

The monotonicity property says that if we increase the suppression probability we can only improve privacy.

THEOREM 1. *Privacy is a monotone property: If $\vec{p}$ preserves $\delta$-privacy then so does any $\vec{q}$ dominating $\vec{p}$.*

A proof can be found in the Appendix A. Furthermore, utility is an anti-monotone property, i.e. we can only decrease utility if we increase suppression probabilities.

OBSERVATION 1. *Utility is an anti-monotone property: Any vector dominating $\vec{p}$ cannot have more utility than $\vec{p}$.*

Our privacy definition has the monotonicity property in common with other definitions such as $k$-anonymity [32] and $\ell$-diversity [22].[7] This monotonicity property allows us to adapt existing efficient search algorithms. We can adapt the greedy approach of Mondrian [21] proposed for $k$-anonymization by starting with the vector $(1, \ldots, 1)$ and gradually reducing the suppression probabilities until reducing any suppression probability further would violate privacy. We end up with a minimal vector. There might be other minimal vectors with more utility, though. To find those we can use the algorithm AlgPR [1] that only relies on the monotonicity of privacy and the anti-monotonicity of utility.

**Privacy.** It is easy to see that the probabilistic check preserves $\delta$-privacy if IsPrivate correctly determines whether the suppression probabilities preserve privacy. IsPrivate is correct because it follows the definition of privacy considering an adversary knowing the probabilistic check and the Markov chain of the user.

LEMMA 1. MaskIt *preserves $\delta$-privacy instantiated with the probabilistic check.*

**Utility.** The following lemma analyzes the utility of using AlgPR in the search of privacy-preserving suppression probabilities that maximize utility.

LEMMA 2. AlgPR *[1] using* IsPrivate *solves the optimization problem from* $\mathsf{initialize}$*: It finds suppression probabilities that maximize utility among all suppression probabilities that preserve $\delta$-privacy.*

## 3.3 Efficiency

The $\mathsf{initialize}$ method of the probabilistic privacy check is expensive. It calls one of the search algorithms, which in turn makes many calls to IsPrivate, each of which can take exponential time in the number of states due to the iteration over possible output sequences. In particular, Mondrian calls IsPrivate $O(Tn\log(d))$ times when using binary search. The number of calls to IsPrivate from AlgPR is $O(Tn\log(d))$ times the number of minimally privacy-preserving vectors plus the number of maximally non-privacy-preserving vectors [1]. We now explore optimizations to improve the running time of IsPrivate to be polynomial. Across calls to IsPrivate, we explain how to re-use partial computations.

**Speeding Up IsPrivate.** To improve the running time of IsPrivate we exploit the independence property of Markov chains (1). Instead of iterating over all possible output sequences $\vec{o}$ in Line (11) in Algorithm 2 to compute the posterior belief of $X_t = s$ given $\vec{o}$, it suffices to consider output subsequences $o_{t_1}, \ldots, o_{t_2}$ of the form $c_i, \perp, \ldots, \perp, c_j$ with $t_1 \leq t \leq t_2$. We replace Line (11) with

```
1: Let $\mathcal{O} = \{o_{t_1}, \perp, \ldots, \perp | t_1 \leq t, o_{t_1} \in \{c_1, \ldots, c_n, \text{“start”}\}\}$
2: $\mathcal{O} \cup= \{o_{t_1}, \perp, \ldots, \perp, o_{t_2} | t_1 \leq t \leq t_2 \wedge$
3:                $o_{t_1}, o_{t_2} \in \{c_1, \ldots, c_n, \text{“start”}, \text{“end”}\}\}$
4: for partial output sequence $\vec{o} \in \mathcal{O}$ do
5:     ...
```

To compute $\Pr[X_t = s | o_{t_1}, \perp, \ldots, \perp, o_{t_2}]$ with $c_i = o_{t_1}$ and $c_j = o_{t_2}$, we adapt Equation (3): We set $\alpha_i(t_1) = 1$, $\alpha_l(t_1) = 0$ (for $l \neq i$) and $\beta_j(t_2) = 1 - p_j$, $\beta_l(t_2) = 0$ (for $l \neq j$). Finally, we can test if $\Pr[o_{t_1}, \ldots, o_{t_2}] > 0$ by testing the following equivalent condition (1) $p_i^{(t_1)}, p_j^{(t_2)} < 1$, (2) $\Pr[X_{t_1} = c_i] > 0$ and (3) $c_j$ is reachable from $c_i$ at time $t_1$ by a path of length $t_2 - t_1$ through states that have non-zero probability of being suppressed.

THEOREM 2. *The running time of the refined* IsPrivate *with Line (11) replaced with Lines (1, 4) is polynomial in the number of contexts and $T$. Its correctness is maintained.*

The proof follows from the independence assumption in our Markov Chain. We can show that in our hidden Markov model $X_t$ is conditionally independent of the output variables other than the ones we iterate over. Two sets of variables $\mathbf{X}, \mathbf{Y}$ are conditionally independent given a third set of variables $\mathbf{Z}$ if $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated* given $\mathbf{Z}$ [8]. This is the case if in any trail (path ignoring the directions of the edges) between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ there exists a node $Z$ such that

- $Z$ has two incoming arrows on the trail $\cdots \to Z \leftarrow \ldots$ and neither $Z$ nor any of its descendants are in $\mathbf{Z}$, or
- $Z$ does not have two incoming arrows on the trail, that is $\cdots \to Z \leftarrow \ldots$, and is in $\mathbf{Z}$

Details can be found in the full version of this paper [29].

**Speeding Up the Search Algorithm.** Both search algorithms, Mondrian and AlgPR, start from high suppression probabilities $\vec{p}$ that

[7] In their case monotonicity is defined over the lattice of full-domain generalizations of the micro-data.

**Algorithm 3** Simulatable privacy check.

```
    procedure initialize($(\delta, M, S)$) return

2:  procedure okayToRelease($\cdot$, $t'$, $\vec{o}$)
        for each possible state $j$ at time $t'$ given $\vec{o}$ do
4:          for each $s \in S$ do
                for $t \in [T]$ do
6:                  Compute prior $\Pr[X_t = s]$ .
                    Compute posterior $\Pr[X_t = s | \langle \vec{o}, c_j \rangle]$.
8:                  if posterior $-$ prior $> \delta$ then
                        return false
10:     return true
```

preserve privacy and use binary search over each probability to see how much it can be decreased without breaching privacy. We can re-use the results from checking privacy of $\vec{p}$ in order to check privacy of $\vec{q}$ dominating $\vec{p}$. If $\vec{p}$ passes the check then so does $\vec{q}$. This fact is already exploited by the two search algorithms. However, we observe that we can get an additional speed-up in these algorithms by caching intermediate results if $\vec{p}$ failed ISPRIVATE.

LEMMA 3. *Let $P(\vec{p})$ (for Passed) denote the set of triplets $\langle t, s, \vec{o} \rangle$ that passed the check, i.e., did not result in false in Line (15) in* ISPRIVATE$(\delta, \vec{p}, S, M)$. *For $\vec{q}$ dominating $\vec{p}$, it suffices to check triplets not in $P(\vec{p})$, i.e., the result of* ISPRIVATE$(\delta, \vec{q}, S, M)$ *will not change if the posterior belief of triples in $P(\vec{p})$ is not computed and not verified to be at most $\delta$ larger than the prior belief.*

# 4. SIMULATABLE PRIVACY CHECK

At current time $t'$ our simulatable check uses Algorithm 3 to decide whether to release or suppress the current state. This decision is made in a *simulatable* way,[8] i.e., only based on information available to the adversary at that time, namely, the Markov chain $M$ and the output sequence $o_1, \ldots, o_{t'-1}$. The current state is ignored. The simulatable check decides to release the current state if for any possible state $c_j$ at time $t'$, releasing $c_j$ does not violate privacy.

**Generation of Possible States.** To compute all possible states at time $t'$ given $\vec{o}$, let $t''$ denote the time of the last output $\neq \perp$. State $c_j$ is a possible state if it is reachable from $o_{t''}$ within $t' - t''$ steps: $(e_{o_{t''}} A^{(t''+1)} \cdots A^{(t')})_j > 0$, where $e_{o_{t''}}$ denotes the $o_{t''}{}^{\text{th}}$ unit vector that has 1 at position $o_{t''}$ and 0 in other positions.

**Details on computing beliefs.** The following proposition describes how an adversary computes his posterior belief.

PROPOSITION 3. *Consider an output sequence $\vec{o} = o_1, \ldots, o_{t'}$ computed by the simulatable check. Consider a time t. Let $t_1$ be the last time before or at t at which a context was released. Let $t_2$ be the earliest time after t at which a context was released. If no such time exists, set $t_2 = T + 1$ and $o_{T+1} =$ "end". The adversary's posterior belief (knowing $M$ and the simulatable check) about a user being in a sensitive context s at time t is*

$$\Pr[X_t = s | X_{t_1} = o_{t_1}, X_{t_2} = o_{t_2}].$$

*where the randomness comes from $M$.*

The proof follows from the simulatability of the check and the independence property of Markov chains (see the full version of this paper for details [29]).

COROLLARY 1. *We can compute the posterior belief of $X_t = s$ given $\vec{o}$ as:*

$$\Pr[X_t = s | \vec{o}] = \frac{\Pr[X_t = s | X_{t_1} = o_{t_1}] \Pr[X_{t_2} = o_{t_2} | X_t = s]}{\Pr[X_{t_2} = o_{t_2} | X_{t_1} = o_{t_1}]}$$

*We use Equation (2) to efficiently compute the transition probability between two states.*

PROOF. We rewrite the posterior belief using the independence property of Markov chains and Proposition 3.

$$\begin{aligned}
&\Pr[X_t = s | \vec{o}] = \Pr[X_t = s | X_{t_1} = o_{t_1}, X_{t_2} = o_{t_2}] \quad \text{By Prop. 3} \\
&= \frac{\Pr[X_t = s, X_{t_1} = o_{t_1}, X_{t_2} = o_{t_2}]}{\Pr[X_{t_1} = o_{t_1}, X_{t_2} = o_{t_2}]} \\
&= \frac{\Pr[X_{t_2} = o_{t_2} | X_{t_1} = o_{t_1}, X_t = s] \Pr[X_t = s, X_{t_1} = o_{t_1}]}{\Pr[X_{t_2} = o_{t_2} | X_{t_1} = o_{t_1}] \Pr[X_{t_1} = o_{t_1}]} \\
&= \frac{\Pr[X_{t_2} = o_{t_2} | X_t = s] \Pr[X_t = s | X_{t_1} = o_{t_1}]}{\Pr[X_{t_2} = o_{t_2} | X_{t_1} = o_{t_1}]} \quad \text{By Eq. 1} \\
&= \frac{\Pr[X_t = s | X_{t_1} = o_{t_1}] \Pr[X_{t_2} = o_{t_2} | X_t = s]}{\Pr[X_{t_2} = o_{t_2} | X_{t_1} = o_{t_1}]}
\end{aligned}$$

This completes the proof. □

**Privacy.** Our check preserves privacy.

THEOREM 3. MASKIT *preserves $\delta$-privacy instantiated with the simulatable check in Algorithm 3.*

See Appendix B for the proof of Theorem 3.

**Utility.** The simulatable check is locally optimal in the sense that if the next state is published despite the indication of the privacy check to suppress it (improving the utility) then there is a chance that future states will inevitably breach privacy.

**Efficiency.** The running time of each call to okayToRelease of the simulatable check is polynomial in the number of contexts and $T$.

We can speed up the check by noticing that many checks in Line (8) in Algorithm 3 are carried out over and over again for consecutive calls of okayToRelease from the system. Some of these checks are redundant. At time $t'$, let $t''$ denote the time of the last output $\neq \perp$ before or at $t'$. For $t < t''$ we have due to Theorem 3 that the output after $t''$ does not affect the posterior belief of $X_t = s$ given $\vec{o}$ for all $s$. Thus in Line (5) in Algorithm 3, it suffices to iterate $t$ over $t'' + 1, \ldots, T$.

# 5. COMPARATIVE ANALYSIS

## 5.1 Utility

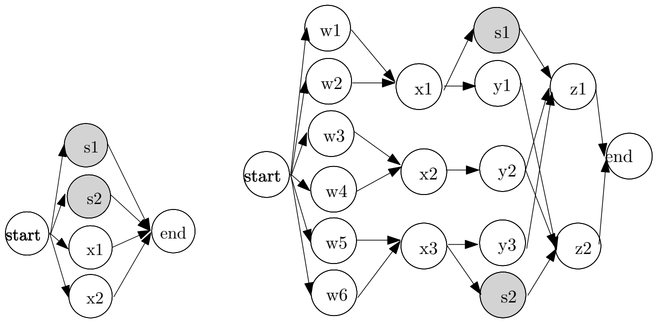
A natural question to ask is which of the two checks (the probabilistic check or the simulatable check) provides more utility. In this section, we study this question from an analytical point of view. The relative benefit of the two checks depends on the user's Markov chain and her sensitive contexts. We give two examples: An example of a Markov chain where the simulatable check performs better than the probabilistic check and an example where the probabilistic check performs better than the simulatable check.

**Probabilistic check is superior.**

EXAMPLE 2. Consider the example Markov chain in Figure 1(a). The transition probabilities are uniform across the outgoing edges of a node. States $s1, s2$ are sensitive. Suppose we want $\delta = 1/4$-privacy. The simulatable check suppresses $X_1$. The probabilistic check, however, only suppresses the sensitive contexts, as this suffices to protect privacy. The prior belief of $X_1 = s_i$ is $1/4$ and the posterior belief given $\perp$ is $1/2$.

This example illustrates a weakness of the simulatable check: It makes the suppression decision without looking at the current state. If there is a chance of currently being in a sensitive state that has a prior belief $< 1 - \delta$ then the simulatable check always suppresses

[8] The notion of simulatability goes back to query auditing [27].



(a) Probabilistic Better (b) Simulatable Better

**Figure 1: Two Markov chains.**

the current state. The probabilistic check considers the current state and in such a case does not necessarily have to suppress it.

**The simulatable check is superior.**

EXAMPLE 3. Consider the example Markov chain in Figure 1(b). States $s1, s2$ are sensitive. Suppose we want $\delta = 1/3$-privacy. The simulatable check outputs one of these sequences: ⟨start, w1, x1 $\perp, \perp$⟩, ⟨start, w2, x1 $\perp, \perp$⟩, ⟨start, w3, x2, y2, z1 ⟩, ⟨start, w3, x2, y2, z2 ⟩, ⟨start, w4, x2, y2, z1 ⟩, ⟨start, w4, x2, y2, z2 ⟩, ⟨start, w5, x3 $\perp, \perp$⟩, ⟨start, w6, x3 $\perp, \perp$⟩. The expected number of released states is $8/3$. For the probabilistic check the utility of all minimally privacy-preserving suppression probabilities is at most $7/3$, e.g., always suppressing s1, y1, z1, s2, y3, and z2 preserves $\delta = 1/3$-privacy and maximizes utility ($7/3$).

Details are omitted for lack of space.

This example illustrates a weakness of the probabilistic check: Its decision ignores the previously released states. It might have to suppress a state because there exists some $\vec{o}$ in Line (11) of Algorithm 2 for which otherwise the posterior belief of some sensitive state is too high. Now, if this $\vec{o}$ is inconsistent with the outputs released so far it might be okay to release the state. For example, if the output released so far is ⟨start, w3⟩ then all the remaining states can be released. The simulatable algorithm makes decisions based on the released states so far and can thus achieve higher utility.

**Hybrid Privacy Check.** How can we analytically determine which one of the two checks is more suitable for a particular user? We explain how to compute the utility of both checks. Then it is easy to pick the better one. Recall from Sec. 2.1 that the utility is defined as the expected number of released states in an output.

For the probabilistic check with suppression probabilities $\vec{p}$ we compute the utility as:

$$\text{utility}^{\mathsf{Prob}}(M) = \sum_{\vec{o}} \Pr[\vec{o}] |\{i | o_i \neq \perp\}| = \sum_{i,t} (1 - p_i^t) \Pr[X_t = c_i]$$

For the simulatable check we introduce a short-hand, $\mathsf{supp}_i(t)$, for the number of suppression symbols immediately following the release of $c_i$ at time $t$.

$$\mathsf{supp}(i,t) = \arg\max_{t_2} t_2 - t \text{ s.t. } \forall t_2' : t < t_2' \leq t_2 :$$
$$\mathsf{okayToRelease}(\,\cdot\,, t_2', \langle o_1, \ldots, o_{t-1} c_i, \perp^{t_2'-t-1}\rangle) == \text{false}$$

where $o_1, \ldots, o_{t-1}$ is some output sequence that is consistent with $o_t = c_i$. If no such sequence exists, then $o_t$ can never be $c_i$ and we define $\mathsf{supp}(i,t)$ to be 0. Using $\mathsf{supp}_i(t)$ we can compute recursively the expected number of suppressions following the release of $X_t = c_i$:

$$E[|\{t_2 | o_{t_2} = \perp, t < t_2 \leq T\}| | o_t = c_i] = \gamma_i(t)$$
$$= \mathsf{supp}_i(t) + \sum_j \Pr[X_{t+\mathsf{supp}_i(t)+1} = c_j | X_t = c_i] \gamma_j(t + \mathsf{supp}_i(t) + 1)$$

Our base case is $\gamma_j(T+1) = 0$ for all $j$. Overall, the utility of the simulatable check is $\text{utility}^{\mathsf{Simulatable}}(M) = T - \gamma_{\text{“start”}}(0)$.

Our hybrid check computes the utility of both the simulatable and the probabilistic check and chooses the one with the higher utility.

THEOREM 4. *The hybrid check correctly chooses the check (simulatable or probabilistic) that provides more utility.*

## 5.2 Efficiency

In the MASKIT system, initialize is computed once offline, while okayToRelease is computed online whenever a new context is extracted. Our privacy checks present different tradeoffs between offline and online computation. The simulatable check does not require any initialization; all its computational overhead is incurred during the filtering. If MASKIT has to go live and create a stream immediately then the simulatable check is the only option. The probabilistic and hybrid checks, conversely, perform most of the computation offline during initialize and are suitable when the offline computation can be performed by a server. With a server we can also speedup the simulatable check by pre-computing $\mathsf{supp}(i,t)$. We experimentally measure the computational costs in Sec. 7.

# 6. LIMITED BACKGROUND KNOWLEDGE

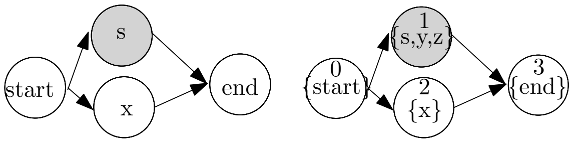
So far, we considered adversaries knowing the *complete Markov chain* $M$ of a user (denoted by $A_M$). Next, we study classes of weaker adversaries with less information $M'$ about $M$ (denoted by $A_{M'}$). We write $M' \preceq M''$ if both $M', M''$ belong to the same class of background knowledge and $M'$ can be extended to $M''$.

**Classes of Adversarial Knowledge.** A *generalized Markov chain* has states labeled with sets of contexts. It captures the knowledge of an adversary who is uncertain about the labels of some states. For example, the adversary might not be able to tell what activity a user is doing in the park (walking or playing frisbee). For other states the adversary might be completely clueless and use a label containing all contexts. The adversary knows that one of these labels in the set is the correct label but is not sure which. Consider a complete chain $M$ with contexts $c_1, \ldots, c_n$. Some of them are declared to be sensitive. A generalized chain has states labeled with sets $c_1', \ldots, c_{n'}'$ in $2^{\{c_1,\ldots,c_n\}}$. Note that two different states at some time $t$ might now have the same set label. To capture the adversary's knowledge that these still are two different states (with two distinct contexts) we change the label to be a pair consisting of the state's ID and the set of contexts. We say a state with ID $a$, context $c_i$ at time $t$ is generalized to a set-labeled state $c_i'$ if $c_i$ is in $c_i'$. $M'$ generalizes $M$ ($M' \preceq M$) if $M'$ can be obtained from $M$ by generalizing each state. We define $\Pi$ to be the generalization from IDs to sets of contexts. Thus, for a weaker adversary the label of a state $a$ in the Markov chain is now $\langle a, \Pi(a)\rangle$.

EXAMPLE 4. For Fig. 2 the transformation is $\Pi(0) = $ {start}, $\Pi(1) = $ {s,y,z}, $\Pi(2) = $ {x}, and $\Pi(3) = $ {end}.

We denote by $X_t'$ the random variable describing the state-ID at time $t$. In relation to $X_t$ we have that $\sum_{i:c_i \in \Pi(a)} \Pr[X_t = c_i | X_t' = a] = 1$ and $\sum_{b:c_i \in \Pi(b)} \Pr[X_t' = b | X_t = c_i] = 1$.

We assume that in the lack of other knowledge the adversary $A_{M'}$ computes her prior belief about the user being in a sensitive context $s$ at time $t$ with ID as $\sum_a \Pr[X_t' = a] \Pr[s | X_t' = a]$.



(a) Complete chain (b) Generalized chain

**Figure 2: Protection against a strong adversary does not imply a protection against a weak adversary.**

A *frequency distribution over states* at each time step captures the knowledge of an adversary unaware of temporal correlations.

A *partial Markov chain combined with frequencies* captures bits and pieces of $M$: For example, an adversary might know that almost every Monday morning some user arrives at work at 10 am. The adversary might further know that the user always stops by the daycare center right after work. This knowledge can be represented as a partial Markov chain with frequency constraints that can be completed to $M$ by adding states and transitions.

**Learning Challenges.** A weak adversary knowing $M'$ can learn and become stronger over time. More background knowledge can be obtained from various sources – one of them being the output sequences themselves. For example, an adversary only knowing the frequencies of states can quickly pick up temporal correlations from the released sequence. We denote by $M'[o]$ the updated model after observing $\vec{o}$. We distinguish weak adversaries that exclusively learn from the output sequence from weak adversaries that also discover other sources of knowledge about $M$.

To protect against weak adversaries who can learn from arbitrary sources and update their model $M'$ we need for all $M''$ ($M' \preceq M'' \preceq M$), for all possible output sequences $\vec{o}$, all times $t$ and all sensitive contexts $s \in S$

$$\Pr_{A_{M''[\vec{o}]}}[X_t = s|\vec{o}] - \Pr_{A_{M''}}[X_t = s] \leq \delta \qquad \text{(LC1)} \qquad (4)$$

Addressing (LC1) is requires some effort because a protection against weak adversaries does not imply a protection against strong adversaries: Consider again the Markov chain from Figure 1(b). It is sufficient to only suppress s1, s2, to protect 1/3-privacy against an adversary knowing only frequencies of states. However, if the adversary learns $M$ then the sensitive states are blatantly leaked.

Moreover, what makes addressing (LC1) tricky is that there are many different ways of reasoning and updating (including random worlds and max entropy [11]). Protecting against only one way of reasoning can lead to serious privacy breaches [19].

To protect against weak adversaries who can learn only from the output sequence to update their model $M'$ we do not need to consider arbitrary $M''$ ($M' \preceq M'' \preceq M$), instead we only need to consider stronger models that can be obtained from updated $M'$ based on an output sequence. We thus need for output sequences $\vec{o'}$, $\vec{o}$, all times $t$ and all sensitive contexts $s \in S$

$$\Pr_{A_{M'[\vec{o'},\vec{o}]}}[X_t = s|\vec{o}] - \Pr_{A_{M'[\vec{o'}]}}[X_t = s] \leq \delta \qquad \text{(LC2)} \qquad (5)$$

Whether it is necessary to address (LC1) or it suffices to address (LC2), is a decision to be made by the data publisher based on what assumptions can be made about the adversaries' knowledge about $M$. Our negative results hold even for (LC2) only, while our positive results address both (LC1) and (LC2).

**No Free Lunch.** It would be nice, if a privacy protection against strong adversaries implies a protection against weak adversaries as is the case for $\ell$-diversity [22, 25]. Unfortunately, this is not the case as the following example illustrates.

EXAMPLE 5. Figure 2(a) shows the complete Markov chain of a user. Our goal is $\delta = 1/4$-privacy. The probabilistic check suppresses s with probability 1 and x with probability 1/3. The simulatable check always suppresses both s and x. Now, consider the weak adversary with a Markov chain with set labels depicted in Figure 2(b). The adversary does not know whether one of the states is a sensitive context labeled with s or a non-sensitive context labeled with either y or z. Suppose he considers these options equally likely. Thus, his prior belief is $\Pr[X_1 = s] = 1/6$. However, upon observing an output sequence containing $\perp$ (either from the probabilistic or the simulatable check) the adversary learns that there must be a sensitive context in the chain. The adversary infers $M$ and his posterior belief about $X_1 = s$ given $\perp$ is $3/4$ for the probabilistic check and $1/2$ for the simulatable check. The increase in belief is drastic ($\geq 1/3$) and violates $\delta = 1/4$-privacy (LC2).

Therefore, we need to find a different way to protect privacy against weaker adversaries that actually addresses (LC2).

**Protection against an Adversary Knowing the Frequency of sensitive contexts.** Adversaries knowing the frequency of sensitive states among other information about $M$ have the same prior belief as the adversary $A_M$.This includes not just adversaries with knowledge about frequency distributions, but also some adversaries with knowledge about a partial or generalized chain. To preserve privacy, we publish the complete Markov chain $M$ together with any output sequence that preserve privacy against $A_M$.

THEOREM 5. *Let $\mathcal{F}(\delta, M, S)$ be a system preserving $\delta$-privacy against strong adversaries knowing the complete Markov chain $M$. Consider a weaker adversary knowing the frequencies of sensitive states among other things about $M$. Using $\mathcal{F}(\delta, M, S)$ to compute a privacy-preserving output sequence $\vec{o}$ and publishing this sequence together with $M$ preserves $\delta$-privacy against adversaries knowing the frequency of sensitive contexts.*

The proof can be found in the full version of this paper [29].

This technique does not necessarily work for weaker adversaries not knowing all frequencies of sensitive contexts. For example, it does not work for Fig. 2(b) due to (LC2).

**Protection against an Adversary Knowing a Set-Labeled Chain.** We can preserve privacy against this class of adversaries using privacy checks designed to protect against the strong adversary. We only need to treat set-labeled states containing a sensitive context as sensitive states and other set-labeled states as non-sensitive states.

THEOREM 6. *Consider a weak adversary knowing $M'$, which is a set-labeled generalization with transformation $\Pi$ of the complete Markov chain $M$ with sensitive contexts $S$. We define $S'$ to be the subset of states in $M'$ that contain at least one sensitive context in $S$. Let $\mu = \max_{s \in S, t \in [T]} |\{a|s \in \Pi(a), \Pr[X'_t = a] > 0\}|$. Let $\mathcal{F}$ be a system that preserves $\delta/\mu$-privacy against adversaries knowing a* complete *Markov chain.*

*$\mathcal{F}(\delta/\mu, M', S')$ given the user input sequence transformed through $\Pi$ preserves $\delta$-privacy against the weaker adversary knowing $M'$.*

The proof is in the full version of this paper [29]. Crucial w.r.t. (LC2) is the fact that output sequences offer no new information to update $M'$ because the adversary can generate output sequences following the same distribution himself based on $M'$.

In order to use Theorem 6 in MASKIT we need to know the background knowledge of each adversarial application. How do we obtain this information? If we ask the applications they might lie about their knowledge. But we can incentivize applications to be truthful by punishing applications reporting incorrect knowledge with de-installation or legal charges. Then rational adversaries have an incentive to be truthful: If they withhold knowledge then their utility decreases while privacy is still preserved. If they make guesses to pretend they have more knowledge then they actually do, they risk being detected and punished.

# 7. EXPERIMENTS

## 7.1 Setup

**Dataset.** We evaluated our system using the Reality Mining dataset.[9] It contains continuous data on daily activities of 100 students and staff at MIT, recorded by Nokia 6600 smartphones over the 2004-2005 academic year [6]. The trace contains various attributes such as a user's location (at granularity of cell towers), proximity to others (through Bluetooth), activities (e.g., making calls, using phone apps), transportation mode (driving, walking, stationary), etc. over different times of the day. We consider 91 users who have at least 1 month of data. The total length of all users' traces combined is 266,200 hours. The average, minimum, and maximum trace length over all users is 122 days, 30 days, and 269 days, respectively. For each user, we train a Markov chain on the first half of his trace; the remaining half is used to for evaluation.

Most of our experiments use the location contexts of all 91 users (as location represents the most complete and fine-grained context in the dataset). The average, minimum, and maximum number of locations per user is 19, 7, and 40, respectively. We also report an experiment with contexts based on users' activities and transportation modes to demonstrate the generality of MASKIT. This information is only available for 23 users.

**Systems.** We compare MASKIT using the simulatable check, the probabilistic check (with a granularity of $d = 10$) and the hybrid check with the naïve approach, called MaskSensitive, which suppresses all sensitive states. For higher values of $d$ we expect the utility but also the computational cost to go up.

**Privacy Configuration.** Unless otherwise stated, we choose $\delta = 0.1$. We experiment with two different ways of choosing sensitive contexts. Unless otherwise stated, we choose sensitive contexts uniformly at random for each user. Alternatively, we choose the home location of a user as the sensitive context.

**Measures.** We measure utility as the fraction of released states in the second half of the trace. We measure privacy breaches as the fraction of sensitive states for which the posterior belief is more than $\delta$ larger than the prior belief. Note, that MASKIT will always assure that there are no privacy breaches. For MaskSensitive an adversary computes his posterior belief as follows: Consider a hidden Markov model defined by $M$ with emission probabilities $b_{i,k}^{(t)} = \Pr[O_t = k | X_t = c_i]$ which is 1 if and only if $k = \perp$ and $c_i \in S$ or $k = i$ and $c_i \notin S$. This hidden Markov model correctly describes the behavior of MaskSensitive. An adversary who knows $M$ and MaskSensitive computes his posterior belief simply as $\Pr[X_t = s|\vec{o}]$ in this hidden Markov model. We can efficiently compute this posterior belief using Equation (3). We say that the privacy of the user's sensitive state $s \in S$ at time $t$ is breached by the output $\vec{o}$ of MaskSensitive if the adversary's posterior belief, $\Pr[X_t = s|\vec{o}]$, is more than $\delta$ larger than his prior belief $\Pr[X_t = s]$. We measure privacy breaches as the number of sensitive states in the user's sequence that are breached divided by the length of the user's sequence.

**Hardware.** Most of our experiments are run on an Intel Xeon 2.33 GHz machine. To measure the overhead of MASKIT when run on a smart phone, we also conduct experiments on a Samsung Focus SGH-i917 phone with the Windows Phone 7.5 operating system.

| Check | initialize | okayToRelease | |
|---|---|---|---|
| | PC | PC | Phone |
| Simulatable | - | 36 ms | 128 ms |
| Probabilistic | 15 min | $< 1$ ms | $< 1$ ms |
| Hybrid | 18 min | $\leq 36$ ms | $\leq 128$ ms |

**Table 1: Average processing times.**

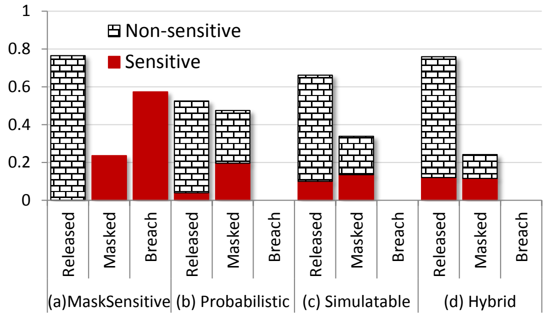


**Figure 3: Comparison of various privacy checks**

## 7.2 Results

**Efficiency.** Before we explore the privacy-utility trade-off, we want to shed light onto the efficiency of various checks. Table 1 shows the average time it takes for MASKIT to initialize the various privacy checks and to filter the trace. Note that on average the suppression decision takes at most 128ms on the phone. If we exclude the slowest 5% of the users this average goes down to 46 ms. This is a negligible overhead compared to the context extraction time of a few up to tens of seconds [2, 26].

The probabilistic and the hybrid check have an expensive initialization even with the speed-up from Section 3 (without which the running time would be exponential). This initialization can be offloaded to a remote server. Overall, in our experiments it seems that the performance of MASKIT is practical for smart phones.

**Privacy Breaches.** Figure 3 reports results from an experiment where we choose three sensitive contexts for each user at random.[10] We report the average fraction of released and suppressed states by various checks. MaskSensitive suppresses sensitive states accounting for 24% of all states. However, this does not prevent an adversary knowing the Markov chain and MaskSensitive from inferring sensitive states: 54% of the suppressed sensitive states still constitute privacy breaches. For these sensitive states the adversary's posterior belief exceeds his prior belief by more than $\delta$. This illustrates the value of having a formal privacy guarantee: With MASKIT no such privacy breaches can happen. Our privacy checks suppress not just sensitive states but also non-sensitive states. (Interestingly, they manage to release some sensitive states without breaching privacy. Those are states with a high prior belief $\geq 1 - \delta$.)

What is the price in terms of utility that we have to pay for a formal privacy guarantee? As Figure 3 shows, the probabilistic check and the simulatable check sacrifice less than 31% and 13% respectively of the utility of MaskSensitive. This appears to be a price well worth the privacy guarantee.

**Hybrid.** From Figure 3(b) and (c), we may get the impression that the simulatable check is superior to the probabilistic check. Despite having a higher average utility across users, the simulatable check is not better for *all* users. Figure 4 shows the utility of both checks for each of the 91 users in our dataset. While for roughly 45% of

[9] http://reality.media.mit.edu/dataset.php.

[10] Recall that states specify time and context. Thus, there are a lot more than three sensitive states in the trace.

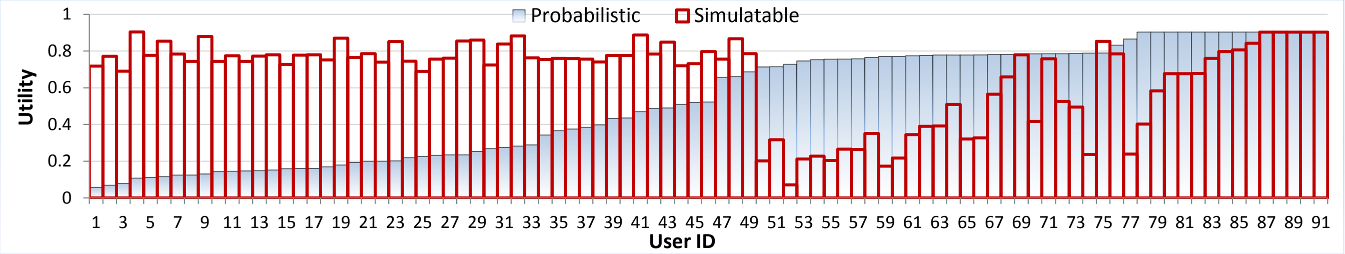


**Figure 4: Utility of two privacy checks for various users. User IDs are sorted based on utility of probabilistic check.**

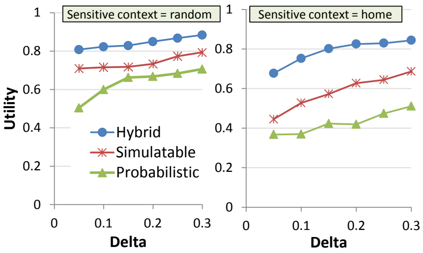


**Figure 5: Privacy-utility tradeoff**

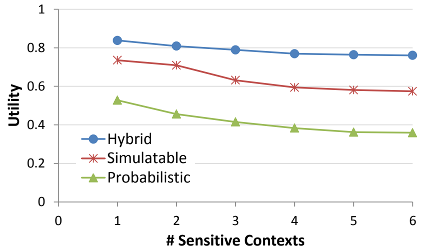


**Figure 6: Utility for Weaker Adversaries. Here $\delta = 0.1 \cdot \#$ Sensitive Contexts.**

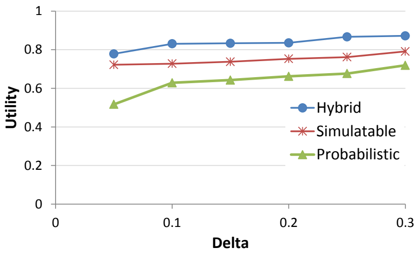


**Figure 7: Activity contexts**

the users the simulatable check is better, for 55% of the users the probabilistic check is better. The goal of the hybrid check is to choose the better check for each user. In our experiment, for 95% of the users the hybrid check picks indeed the check suppressing fewer states in the trace. The hybrid check makes mistakes only for users for which the fraction of suppressed states in the trace differs significantly from the expected fraction of suppressions. Note that by the Law of Large Numbers, for longer traces the fraction of suppressions will be more concentrated around the expectation thus decreasing the number of mistakes of the hybrid. Overall, the hybrid achieves an average utility of 75.8% (see Figure 3(d)) which is much higher than both the utility of the probabilistic and the simulatable check and almost matches that of MaskSensitive (76.4%).

In fact, our hybrid checks provides more utility than MaskSensitive when we increase the number of sensitive states (we omit this experiment due to space constraints). Here, unlike MaskSensitive our hybrid check releases some of the sensitive states without breaching privacy and suppresses fewer states in total.

MaskSensitive provides the highest utility relative to the hybrid check when there is only one sensitive context per user; nevertheless, our hybrid check provides a utility of 84% in this case, which is within 7% of MaskSensitive's utility (91%). This shows that in our experiments the price for a provable privacy guarantee is minor.

**Privacy-Utility Tradeoff.** We also vary the target privacy level by varying the value of $\delta$. We conduct two sets of experiments: In the first set, we choose one sensitive context for a user at random; in the second set, we choose the sensitive context for a user to be his home. When we increase privacy (by decreasing $\delta$), we expect utility to decrease. As we can see from Figure 5, for both sets of experiments, the overall decrease in utility is small. This implies that in our experiments we can afford strong privacy guarantees (by choosing a smaller value of $\delta$) without sacrificing too much utility.

**Limited Background Knowledge.** As explained in Section 6, we can protect against weaker adversaries knowing the frequencies of sensitive states simply by releasing a user's Markov chain along with any output sequence that protects privacy against the strong adversary. This does not affect utility and we obtain the same privacy-utility tradeoff. To protect against a weaker adversary knowing only a generalized Markov chain with set-labeled states, however, we expect the utility to decrease. This is because our learning challenges require that we provision our system to protect against the adversary as he learns and becomes stronger.

Figure 6 measures the utility when protecting against adversaries of varying strength. As the adversary knows less about the Markov chain, the states have an increasing number of labels. This only affects the utility if a non-sensitive state in $M$ now is labeled with a set of contexts including a sensitive context. In that case, our adapted privacy checks will treat it as sensitive according to Theorem 6. This results in a decrease in utility, but also in decrease of privacy, i.e., the effective privacy guarantee degrades to $0.1\cdot$ the number of sensitive contexts.

For this experiment, we picked a subset of 25 users with a number of contexts between 15 and 20. The effect of increased uncertainty will be more (less, respectively) drastic for users with fewer (more, respectively) contexts. Figure 6 shows the effect on utility as the adversary knows less about $M$, i.e., as the number of *potentially* sensitive states grows. While the utility decreases for all three checks, the rates of decrease differ; the hybrid's utility decreases slowest.

**Beyond Location.** The experiments so far used location as a context. To show that MaskIt can operate with other types of context, we now consider user contexts that are combinations of the user's activities (making a phone call, sending an sms, using an application on the phone) and his transportation mode (sitting, walking, riding a bus or car).[11] As in Figure 5(left), we choose a single sensitive context for each user at random. Figure 7 shows the privacy-utility tradeoff for the activity contexts—the results are very similar to the results for location contexts.

## 8. RELATED WORK

Prior work has considered preserving privacy while releasing a user's location in a location-based service (LBS). Many existing privacy techniques focus on the "single shot" scenario [17]. Unlike MaskIt, these techniques do not protect privacy against adversaries knowing temporal correlations. Anonymity-based techniques aim to hide the identity of a user issuing a query specifying his location, by replacing user's exact location with a broader region containing at least $k$ users [12, 32, 31]. However, $k-$anonymity

[11] The transportation mode is inferred from survey responses.

does not readily imply privacy, e.g., $k$ users can be in the same sensitive location. As we discussed before, MaskSensitive-like naïve approach, which masks sensitive locations [34] or sensitive patterns [14], cannot guarantee privacy.

There has been work to protect against adversaries aware of some temporal correlations [5, 9, 10, 13, 28]. Gruteser and Liu [13] consider an adversary applying linear interpolation to infer suppressed locations. They introduce uncertainty about sensitive locations by creating zones so that each zone has multiple sensitive locations. This approach does not prevent privacy breaches completely but reduces them in comparison the naïve approach. Cheng et al. [5] consider an adversary knowing the maximum velocity of users. Given two consecutive cloaked regions of a user the adversary can exclude points in the second region that are unreachable from any point in the first one. They protect against this attack but not against adversaries also knowing the system. This work is improved by Ghinita et al. [9] using spatial cloaking and introducing delays. However, the delay can leak information about the user's exact location and is thus vulnerable to an attack from an adversary knowing a little bit about the distribution of the time between consecutive queries: If the delay is just long enough to make every point in the second region accessible from every point in the first region then it is likely that the second region has been artificially delayed. Parate and Miklau release not location but communication traces [30]. A trace is transformed so that the number of possible traces consistent with the transformed trace is maximized subject to a constraint on utility. This technique does not provide a semantic privacy guarantee. In summary, the work by [5, 9, 13, 30] does not provably protect privacy against adversaries knowing the system and temporal correlations beyond the max velocity.

Several recent cryptographic protocols can provably provide privacy against these adversaries [10, 28]. However, these protocols can only be used to answer nearest neighbor queries [10] and find close-by friends [28]. They cannot be used to release a privacy-preserving stream of contexts.

To the best of our knowledge, MASKIT is the first system releasing context streams that protects privacy against very strong adversaries knowing the system and temporal correlations in the form of a Markov chain that go far beyond the max velocity. Moreover, our contexts are not limited to location, but can include the social state and other activities. This enables more powerful personalizations.

# 9. CONCLUSIONS

We addressed the problem of privately releasing user context streams. Our system, MASKIT, employs a privacy check that decides whether to release or suppress the current user context. We presented two privacy checks that provably guarantee privacy against powerful adversaries knowing the system and temporal correlations in the stream. They differ, though, in their utility for a user and our hybrid check determines the one with the higher utility. To also protect against weaker adversaries, who can learn and become stronger, we adapted our privacy checks. Our experimental evaluation on real context traces demonstrates that we do not have to sacrifice much utility in order to guarantee privacy.

**Acknowledgments.** This material is based upon work supported by the National Science Foundation under Grants IIS-1012593 and IIS-0911036. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank the authors of [1] for providing us the code of the search algorithm used by the probabilistic check. Gabriel Bender, Gabriel Ghinita, Moritz Hardt, Michael Hay, Yeye He, and Guozhang Wang have provided helpful comments.

# 10. REFERENCES

[1] Arvind Arasu, Michaela Götz, and Raghav Kaushik. On active learning of record matching packages. In *SIGMOD*, 2010.

[2] Gerald Bieber, Jörg Voskamp, and Bodo Urban. Activity recognition for everyday life on mobile phones. In *HCI*, 2009.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[4] Soumen Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *WWW*, 2007.

[5] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Privacy Enhancing Technologies*, 2006.

[6] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *Proceedings of the National Academy of Sciences (PNAS)*, 106:15274–15278, 2009.

[7] William Enck, Peter Gilbert, Byung gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *OSDI*, 2010.

[8] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2007.

[9] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. Preventing velocity-based linkage attacks in location-aware applications. In *GIS*, 2009.

[10] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD*, 2008.

[11] Adam J. Grove, Joseph Y. Halpern, and Daphne Koller. Random worlds and maximum entropy. *J. Artif. Int. Res.*, 2:33–88, 1994.

[12] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[13] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 2:28–34, 2004.

[14] Yeye He, Siddharth Barman, Di Wang, and Jeffrey F. Naughton. On the complexity of privacy-preserving complex event processing. In *PODS*, 2011.

[15] E. Kim S. Helal and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9(1):48–53, 2010.

[16] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW*, 2003.

[17] Christian S. Jensen, Hua Lu, and Man Lung Yiu. *Location Privacy Techniques in Client-Server Architectures*, pages 31–58. Springer-Verlag, 2009.

[18] Patrick Gage Kelley, Michael Benisch, Lorrie Faith Cranor, and Norman M. Sadeh. When are users comfortable sharing locations with advertisers? In *CHI*, 2011.

[19] Daniel Kifer. Attacks on privacy and definetti's theorem. In *SIGMOD*, 2009.

[20] Alexander Kuenzer, Christopher Schlick, Frank Ohmann, Ludger Schmidt, and Holger Luczak. An empirical study of dynamic bayesian networks for user modeling. In *UM*, 2001.

[21] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, 2006.

[22] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. $\ell$-diversity: Privacy beyond $k$-anonymity. *TKDD*, 1(1), 2007.

[23] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[24] Andrea Mannini and Angelo Maria Sabatini. Accelerometry-based classification of human activities using markov modeling. *Computational Intelligence and Neuroscience*, 2011.

[25] David J. Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, and Joseph Y. Halpern. Worst case background knowledge for privacy preserving data publishing. In *ICDE*, 2007.

[26] Emiliano Miluzzo, Cory T. Cornelius, Ashwin Ramaswamy, Tanzeem Choudhury, Zhigang Liu, and Andrew T. Campbell. Darwin

phones: the evolution of sensing and inference on mobile phones. In *MobiSys*, 2010.

[27] Shubha U. Nabar, Bhaskara Marthi, Krishnaram Kenthapadi, Nina Mishra, and Rajeev Motwani. Towards robustness in query auditing. In *VLDB*, 2006.

[28] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, and Dan Boneh. Location privacy via private proximity testing. In *NDSS*, 2011.

[29] Michaela Götz, Suman Nath, and Johannes Gehrke. MaskIt: Privately releasing user context streams for personalized mobile applications. Technical Report MSR-TR-2012-29, Microsoft Research, March 2012.

[30] Abhinav Parate and Gerome Miklau. A framework for safely publishing communication traces. In *CIKM*, 2009.

[31] Linda Pareschi, Daniele Riboni, Alessandra Agostini, and Claudio Bettini. Composition and generalization of context data for privacy preservation. In *PerComm*, 2008.

[32] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, CMU, SRI, 1998.

[33] Eran Toch, Justin Cranshaw, Paul Hankes Drielsma, Janice Y. Tsai, Patrick Gage Kelley, James Springfield, Lorrie Cranor, Jason Hong, and Norman Sadeh. Empirical models of privacy in location sharing. In *Ubicomp*, 2010.

[34] Janice Tsai, Patrick G Kelley, Lorrie F Cranor, and Norman Sadeh. Location sharing technologies: Privacy risks and controls. *I/S: A Journal of Law and Policy for the Information Societ*, 6(2), 2010.

[35] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In *VLDB*, 2007.

[36] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, Philip S. Yu, and Jian Pei. Can the utility of anonymized data be used for privacy breaches? *TKDD*, 5(3), 2011.

[37] Xiaokui Xiao, Yufei Tao, and Nick Koudas. Transparent anonymization: Thwarting adversaries who know the algorithm. *ACM Trans. Database Syst.*, 35(2), 2010.

# APPENDIX

## A. MONOTONICITY PROPERTY OF PROBABILISTIC CHECK

Proof of Theorem 1. Consider vectors $\vec{p}, \vec{q}$ so that $\vec{q}$ dominates $\vec{p}$ and is larger by $\epsilon$ in exactly one dimension: $q_i^{t'} = p_i^{t'} + \epsilon$. Suppose that $\vec{p}$ preserves $\delta$-privacy. To simplify exposition, we introduce a notation. With two different suppression probabilities $\vec{p}, \vec{q}$, we use a subscript to specify which one is used in the computation of a particular probability. For example, we write $\Pr_{\vec{p}}[X_t = s|\vec{o}]$. We might change one of the values $p_j^t$ to $v$ and write $\vec{p}[p_j^t = v]$ to denote the new suppression probabilities.

In order to prove that also $\vec{q}$ preserves $\delta$-privacy we need to show that the maximum difference (over sensitive states $s$, time $t$, outputs $\vec{o}$) between posterior and prior belief does not increase when going from $\vec{p}$ to $\vec{q}$. Fix a sensitive state $s$ and a time $t$. It suffices to show that the maximum (over outputs $\vec{o}$) of the posterior belief does not increase. In particular, we show that for all $\vec{o}$ either $\Pr_{\vec{p}}[X_t = s|\vec{o}] \geq \Pr_{\vec{q}}[X_t = s|\vec{o}]$ or if that is not the case, then there exists an $\vec{o'}$ such that $\Pr_{\vec{p}}[X_t = s|\vec{o'}] = \Pr_{\vec{q}}[X_t = s|\vec{o'}] \geq \Pr_{\vec{q}}[X_t = s|\vec{o}]$. We consider three cases: Either $o_{t'} = c_i$ or $o_{t'} = c_j$ (for some $j \neq i$) or $o_{t'} = \perp$.

$\underline{o_{t'} = c_j \text{ for } j \neq i}$: Recall that according to Proposition 2 the posterior belief of $X_t = s|\vec{o}$ is computed in the hidden Markov model defined by $M$ and the emission matrices. Changing $p_i^{t'}$ only changes $b_{i,\perp}^{t'}$ and $b_{i,i}^{t'}$. All other emission probabilities remain unchanged. As we can see from Equation (3) and the definition of $\alpha$ and $\beta$ the changed emission probabilities are not part of the computation of $\Pr[X_t = s|\vec{o}]$. Thus, we have that $\Pr_{\vec{p}}[X_t = s|\vec{o}] = \Pr_{\vec{q}}[X_t = s|\vec{o}]$.

$\underline{o_{t'} = c_i}$: In that case increasing $p_i^{t'}$ has no effect on the probability of being in a sensitive state given $\vec{o}$.

$$\Pr_{\vec{p}}[X_t = s|\vec{o}] = \frac{\Pr_{\vec{p}}[X_t = s, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}]} = \frac{p_i^{t'} \Pr_{\vec{p}[p_i^{t'}=1]}[X_t = s, \vec{o}]}{p_i^{t'} \Pr_{\vec{p},[p_i^{t'}=1]=1}[\vec{o}]}$$

$$= \frac{\Pr_{\vec{q}[q_i^t=1]}[X_t = s, \vec{o}]}{\Pr_{\vec{q}[q_i^t=1]}[\vec{o}]} = \frac{q_i^{t'} \Pr_{\vec{q}[q_i^{t'}=1]}[X_t = s, \vec{o}]}{q_i^{t'} \Pr_{\vec{q}[q_i^{t'}=1]}[\vec{o}]} = \Pr_{\vec{q}}[X_t = s|\vec{o}]$$

$\underline{o_{t'} = \perp}$: If $\Pr_{\vec{p}}[X_t = s|\vec{o}] \geq \Pr_{\vec{q}}[X_t = s|\vec{o}]$ we are done. So consider the case where $\kappa \Pr_{\vec{p}}[X_t = s|\vec{o}] = \Pr_{\vec{q}}[X_t = s|\vec{o}]$ for some $\kappa > 1$. To complete the proof, we show that for $\vec{o'} = \vec{o}$ except for $o'_{t'} = i$: $\Pr_{\vec{p}}[X_t = s|\vec{o'}] = \Pr_{\vec{q}}[X_t = s|\vec{o'}] \geq \Pr_{\vec{q}}[X_t = s|\vec{o}]$. The first equality is a result of the calculations above. Thus to suffices to show the following claim:

CLAIM 1. $\Pr_{\vec{p}}[X_t = s|\vec{o'}] \geq \Pr_{\vec{q}}[X_t = s|\vec{o}]$.

We have that

$$\Pr_{\vec{q}}[X_t = s|\vec{o}] = \kappa \cdot \Pr_{\vec{p}}[X_t = s|\vec{o}]$$

$$\Leftrightarrow \frac{\Pr_{\vec{p}}[X_t = s, \vec{o}] + \epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_t = s, X_{t'} = i, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}] + \epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]}$$

$$= \kappa \cdot \frac{\Pr_{\vec{p}}[X_t = s, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}]}$$

$$\Leftrightarrow 1 + \frac{\epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_t = s, X_{t'} = i, \vec{o}]}{\Pr_{\vec{p}}[X_t = s, \vec{o}]}$$

$$= \kappa \left(1 + \frac{\epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}]}\right)$$

$$\Leftrightarrow \frac{\Pr_{\vec{p}[p_i^{t'}=1]}[X_t = s, X_{t'} = i, \vec{o}]}{\Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]}$$

$$= \frac{(\kappa - 1) \Pr_{\vec{p}}[X_t = s, \vec{o}]}{\epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]} + \kappa \frac{\Pr_{\vec{p}}[X_t = s, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}]}$$

We use this to prove the Claim (1). We rewrite the left-hand side.

$$\frac{\Pr_{\vec{p}}[X_t = s, \vec{o'}]}{\Pr_{\vec{p}}[\vec{o'}]} = \frac{\Pr_{\vec{p}[p_i^{t'}=1]}[X_t = s, X_{t'} = i, \vec{o}]}{\Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]}$$

$$= \frac{(\kappa - 1) \Pr_{\vec{p}}[X_t = s, \vec{o}]}{\epsilon \Pr_{\vec{p}[p_i^{t'}=1]}[X_{t'} = i, \vec{o}]} + \kappa \frac{\Pr_{\vec{p}}[X_t = s, \vec{o}]}{\Pr_{\vec{p}}[\vec{o}]}$$

$$\geq \kappa \Pr_{\vec{p}}[X_t = s|\vec{o}] = \Pr_{\vec{q}}[X_t = s|\vec{o}]$$

We arrive at the right-hand side of Claim (1) completing the proof.

## B. PRIVACY GUARANTEE OF SIMULATABLE CHECK

Proof of Theorem 3. Consider a sensitive context $s$ and a time $t$ and an output $\vec{o}$ produced by the simulatable check. We argue that the posterior belief of $X_t = s$ given $\vec{o}$ is at most $\delta$ larger than the prior belief. Theorem 3 states that the posterior belief of $X_t = s$ given $\vec{o}$ is equal to $\Pr[X_t = s|X_{t_1} = o_{t_1}, X_{t_2} = o_{t_2}]$, where $t_1, t_2$ denote the time of the two released states closest to $t$. (If no such time $t_2$ exists, we set $t_2 = T + 1$ and $o_{t_2} =$"end".) We distinguish two cases based on whether $t_2 = T + 1$.

$\underline{t_2 < T + 1}$: In this case, the check verified that this posterior belief is not too large before releasing $o_{t_2}$ at time $t_2$.

$\underline{t_2 = T + 1}$: In this case, consider the decision to release $t_1$: If $t_1 = 0$ then the prior belief is equal to the posterior belief. Otherwise, when $o_{t_1}$ was released the check verified that the posterior belief of $X_t = s$ given $X_{t_1} = o_{t_1}$ and $X_{T+1} =$"end" is not too large.