Pervasive Context Sharing in MAGPIE: Adaptive Trust-Based Privacy Protection

Chenguang Liu and Christine Julien

Department of Electrical & Computer Engineering
The University of Texas at Austin
Austin, Texas USA
{liuchg,c.julien}@utexas.edu

Abstract. Today's mobile and pervasive computing devices are embedded with increasingly powerful sensing capabilities that enable them to provide exceptional spatio-temporal context acquisition that is not possible with traditional static sensor networks alone. As a result, enabling these devices to *share* context information with one another has a great potential for enabling mobile users to exploit the nearby cyber and physical environments in *participatory* or *human-centric* computing. However, because these mobile devices are owned by and sense information about *individuals*, sharing the acquired context raises significant privacy concerns. In this paper, we define MAGPIE, which implements an alternative to existing *all-or-nothing* sharing solutions. MAGPIE integrates a decentralized *context-dependent* and adaptive trust scheme with a privacy preserving sharing mechanism to evaluate the risk of disclosing potentially private data. The proposed method uses this assessment to dynamically determine the sharing strategy and the *quality* of the context shared. Conceptually, MAGPIE allows devices to actively obfuscate context information so that sharing is still useful but does not breach user privacy. To our knowledge this is the first work to take both trust relationships and users' individual privacy sensitivities into account to balance sharing and privacy preservation. We describe MAGPIE and then evaluate it in a series of application-oriented experiments running on the Opportunistic Network Environment (ONE) simulator.

Key words: Context sharing, privacy preserving, adaptive trust

1 Introduction

With the rapid development of the *Internet of Things* (IoT), everyday consumer devices have become more connected to one another [1]. This offers a chance for these devices to collaborate, which brings opportunities for new applications that can exploit the surrounding environment, especially when these devices are carried by people. By sharing local contextual information, mobile devices can help us to avoid traffic on the road (e.g., Waze[1]), improve recreational sports

[1] https://www.waze.com/

experiences (e.g., BikeNet [2]), and even monitor air pollution (e.g., P-Sense [3] and Citisense [4]). With this shift in the usage comes a shift in how pervasive computing applications view context beyond simple *egocentric* views [5], collected by a single device or user for consumption by a that device or user. The collective or cumulative feature of a set of shared contexts is increasingly valued because of applications in *participatory* or *human-centric sensing* [6]. However, sharing the context information sensed by a user's personal mobile device poses a significant threat to the user's privacy if it is not under proper control.

Given the privacy concerns raised when collecting and sharing information using personal devices, there has been substantial research on two related topics: dynamic trust management and schemes to obfuscate and protect potentially personal data. The goal of dynamic trust management in pervasive computing is to select generally reliable candidates with which to interact (i.e., share information) based on previous experience or general recommendations [7, 8, 9]. On the privacy preservation side, the focus is identifying and perturbing sensitive information to protect an individual from being identified [10, 11, 12]. In isolation, neither of these is effective enough for a context-sharing scenario like *Social Cycling* [2], where the mobile devices carried by a group of cyclists should be able to efficiently provide context data to other participants in the group in order to share up-to-date and reliable information about the availability (and potential availability) of shared bicycles. Such an application requires sharing individual's location traces with other users; most people are not eager to share detailed raw information about their spatiotemporal trajectories with just anyone.

We introduce Magpie, a trust-adaptive and privacy-preserving approach for pervasive context sharing applications in which mobile and heterogeneous sensor-equipped devices *opportunistically* work together to increase awareness of the environment. Magpie facilitates device-to-device context sharing (i.e., without assistance from an infrastructure), as opposed to an approach that relies on dedicated sensors deployed in the environment that are often designed to intentionally provide context information for users without raising privacy concerns. In Magpie, the interaction experience that comes from sharing context information also serves as evidence for later trust establishment. Magpie provides an alternative to traditional *all-or-nothing* sharing approaches by potentially disclosing some obfuscated but still useful context information. A key challenge is to address the privacy concerns of the participants about whom the context is collected while ensuring that the *quality* of context shared is sufficient. Therefore our approach leverages trust relationships established among pervasive computing participants and privacy sensitivities of the individuals together to design the obfuscating process into our context sharing mechanism. Magpie also utilizes *context similarity factors* and *situational trust* to fit the context sharing behavior to the situations of the pervasive computing devices and their users.

To our knowledge this is the first work to use both trust relationships and an individual's privacy sensitivities to estimate the *risk* of context sharing; we use this risk to dynamically select sharing strategies and to affect the *quality* of shared context. To evaluate Magpie, we perform application-oriented exper-

iments on the Opportunistic Network Environment (ONE) simulator [13]. We evaluate the effectiveness of our trust establishment scheme and privacy protection by analyzing the changes in participation in sharing activities as well as the empirical error percentage in the information shared. In Section 2 we outline the related works addressing privacy and trust issues in pervasive computing. The overview, design, and implementation details of Magpie are presented in Section 3, followed by the evaluation of our work in Section 4.

2 Related Work

By sharing context information acquired by a *set* of devices, a group of opportunistically interconnected devices with disparate sensing capabilities is able to be more adaptive to its nearby physical and cyber environments [14, 15]. Magpie is motivated by this new type of application, and we aim to provide a balance between preserving privacy and facilitating context sharing participation. Before describing our approach in detail, we overview related projects establishing trust among distributed pervasive computing participants, addressing privacy in pervasive computing, and supporting context sharing in these environments.

Establishing trust among pervasive computing participants. Users distinguish their expectations of their systems into *familiarity*, *confidence*, and *trust* [16], where the latter uniquely depends not on actual or inherent danger but on the user's perceived *risk*. These perceptions emerge as a part of decision and action. With respect to expectations for sharing context in pervasive computing, trust is fundamental for establishing the sharing relationship between the participants and for selecting the means of the sharing behavior.

Our setting demands a decentralized approach to trust management that can operate without persistent connectivity to the Internet infrastructure. Three branches of decentralized trust management systems exist in the literature: (1) approaches that rely on encounters with trusted third parties and focus largely on cryptographic issues in the authorization process [17]; (2) reputation mechanisms that use social control to store and disseminate reputation information [18, 19, 20]; and (3) purely decentralized trust management systems that establish trust relationships between the devices in pervasive environments based only on inter-device interactions [7, 9]. Because we do not wish to limit the applicability of our approach, we target situations like the latter. However, these existing trust schemes are not tied to determining when and how to share context information, so they require some updating to address the needs of Magpie.

Privacy preservation in pervasive computing. On the other hand, protecting privacy of users' personal information is also a well-studied area. One of the widely accepted works is to use *k-anonymity* [21] for statistical disclosure control; *k-anonymity* aims to render a particular piece of data indistinguishable among the aggregation of $k-1$ other pieces. These approaches are commonly used to protect individuals from being identified given a large amount of aggregated information like medical record data. Approaches that are perhaps more appropriate to pervasive computing environments are based on the idea of adding

noise to personal data on the *client-side* to ensure individual privacy. These systems then use community-wide reconstruction techniques to restore knowledge about a shared group context [11, 22]. Even these latter approaches assume one or more dedicated and honest aggregators within the network, which is limiting for general-purpose pervasive computing environments.

Distributed differential privacy methods [10], derived from classical differential privacy [12], can be applied to allow applications to learn only some important statistics but no additional information and thus satisfy privacy guarantees. These approaches generally require a very large number of data items to be able to provide reasonable privacy while maintaining correct information. Therefore differential privacy based approaches do not suit our needs for sharing context among sparsely connected devices.

More recently, efforts related to data preprocessing in smart grids has demonstrated the ability to obfuscate individual users' behaviors [23]. MAGPIE is inspired by the latter and by distributed differential privacy, but we introduce new noise models to eliminate the characteristics of individual data without losing its inherent meaning. We do assume the availability of a context specific privacy sensitivity manager [24, 25] on each user's device. This privacy manager is able to offer a quantified sensitivity value $\varepsilon \in (0, 1)$ for each type of context, which provides an individualized perception of how private the particular context type is. For instance, a particular user may deem his location context information to be highly private while his ambient sound level context may be less private.

Sharing context. MAGPIE provides capabilities that allow mobile devices to share their sensed context with one another. The potential applications of this work include systems like BikeNet [2] or P-Sense [3], or generally mobile and pervasive computing applications that take advantage of directly sharing context information (e.g., workout companion applications like "Run with a buddy"). Our approach can also be used to extend participatory sensing systems (e.g., a crowd-sourced transit information system [26] or CarTel [27]), especially the ones collaborating in a device-to-device fashion [28, 29, 30]. MAGPIE is primarily motivated by our own previous work on the Grapevine context framework [5], which was developed for succinctly summarizing and efficiently sharing context information in pervasive computing environments.

3 MAGPIE: Adaptive Trust- & Privacy-Based Context Sharing

We consider a network of users with smart devices that are connected to one another by an opportunistic mobile network of device-to-device links[2]. Users' applications collect and act on context information that describes the user's state and situation; this information comes both from the user's own device and through opportunistic sharing with connected devices of other users. For

[2] We use "device" and "user" interchangeably because we assume that every participant is associated with a single device through which he collaborates.

example cyclists can increase energy efficiency or data accuracy if their devices are wisely and effectively sharing information about the riders' trips [2] (e.g., sharing compass information with users in a traveling group whose devices lack that particular sensing capability or taking turns collecting motion statistics to distribute sensing costs). We assume devices operate under a shared context ontology, i.e., we assume that there is a well known set of context types and that the names of these types are shared among all of the participants *a priori*.

We introduce Magpie, which facilitates context sharing activities to make it possible for users to adjust their behavior based on the sensed context while maintaining the privacy of the users about whom the context information is collected. Consider a classic context-awareness scenario [31] in which smart devices are able to adjust themselves and thereby the ambient environment by collecting and actuating on high-level situational knowledge (e.g., the start of a meeting or a social event like a coffee break) inferred from the shared context acquired from multiple devices. Magpie has two key components: adaptive trust evaluation and privacy preserving context sharing.

A key principle of Magpie is that users share multiple types of context information with several other users. For this reason, both the trust evaluation scheme and the privacy sensitivity are *context-dependent*. This reflects the fact that, simply because a coordinating partner is a good source for one type of context information (e.g., local weather) does not necessarily imply he is trustworthy with some particularly personal data (e.g., raw location). Magpie assumes that each user is associated with an individualized specification of their privacy sensitivities for each type of context information shared and maintained by a privacy sensitivity manager (see Section 2). These sensitivity values range over $(0, 1]$, where larger values indicate higher privacy requirements.

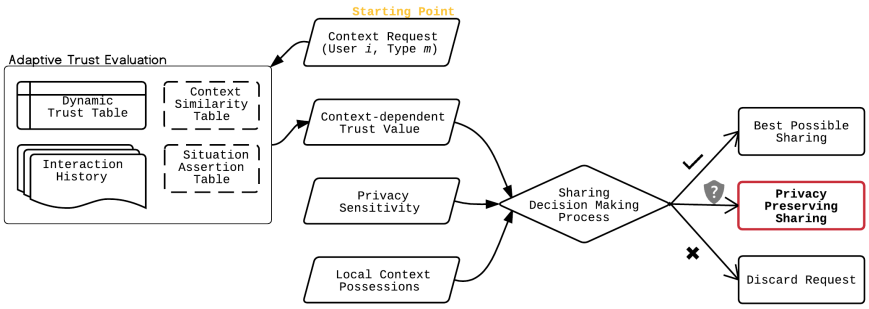

Fig. 1: System Overview

Fig. 1 shows an overview of Magpie, specifically in the process of responding to a neighboring device's request for a piece of context. Upon receiving a request from user u_i for a specific type of context information, m (top center of the

figure), the request passes to the adaptive trust management module to evaluate how trustworthy u_i is regarding the type m. Intuitively, the device determines whether u_i is trustworthy enough to share the raw context information with. If not, the device needs to determine whether it is possible to share *any* knowledge about this context type with u_i, e.g., in an obfuscated form. The quantified result $\tau_{i,m}$ of trust evaluation is considered, together with the user's privacy sensitivity for the context type m (ε_m) and the local context possession C_m, to assess the potential *risk* of sharing the requested context information with u_i. The context sharing module uses this risk to select a sharing strategy that maximizes the possibility of participation while keeping any potential privacy breach under control. Fig. 1 shows three possibilities: (1) there is no risk, so the request can be fully satisfied with the raw data; (2) there is some mitigable risk, and MAGPIE shares some obfuscated context data; and (3) the risk is intolerable, and the request is discarded. The rest of this section provides the details of MAGPIE's two essential components.

3.1 Adaptive Trust Management

In MAGPIE, the sharing decision is made based on several factors as described earlier, but the foundation is an established level of trust between the recipient of the request and the peer initiating the request. MAGPIE makes it possible to potentially disclose some obfuscated but still useful context information, even if the requesting peer is not fully trustworthy. Therefore the trust a potential sharer of context information has in the requesting peer not only partially determines which option to take, but also relates to how useful the information will be. As such, having an expressive and effective mechanism to dynamically evaluate the trust that a participant has in some requesting peer is essential to MAGPIE. We define trust (as perceived by a particular user u_i) as follows:

Definition 1. *Trust. For a given user u_i, the value of Trust, $\tau^i_{j,m} \in (0,1)$ indicates to which extent a context requester u_j can be trusted with respect to a particular context type, m.*

We build on the wealth of mathematical models of trust and incorporate decentralization, personalization, and specificity to the type of context information being shared. To start, we use the *Pervasive Trust Management* model [7] based on Luhmman's idea [16] as a foundation. This definition of trust relies on a log of user i's satisfaction (or dissatisfaction) in his historical interaction experience $a^i_{j,k}$ with a particular peer u_j. To account for these dynamics, we extend the above definition of trust with a notion of timestep. In this extended model, user i's trust in user j for context type m after interaction k is defined as:

$$\tau^i_{j,m,k} = \begin{cases} \tau^i_{j,m,k-1} + \omega \cdot V_{a^i_{j,k}}(1 - \tau_{j,m,k-1}) & V_{a_k} > 0 \\ \tau^i_{i,m,k-1}(1 - \omega + \omega \cdot V_{a^i_{j,k}}) & else \end{cases} \tag{1}$$

where $V_{a^i_{j,k}}$ is the product of the satisfaction (a^+) and dissatisfaction (a^-) of the past behaviors. Satisfaction and dissatisfaction can be measured in a variety

of ways. In Magpie, we count satisfaction (a^+) as the percentage of times in which a request from i to j for context type m resulted in a response and dissatisfaction as the percentage of times in which such an interaction did not result in any response. This is a simple scheme that could easily be extended, but this is not the primary focus of this work. The updated trust value is also weighted according to a user- or system-defined weight (ω).

In Magpie, the actions through which users can learn about others' trustworthiness involve context requesting and sharing, thus it is natural to make $V_{a^i_{j,k}}$ also be context dependent. Specifically, with regard to context type m, a $V_{a^i_{j,m,k}}$ can be calculated independently for each type of context that may be requested (and context-specific satisfaction measures) using the equation below[3]:

$$V_{a^i_{j,m,k}} = \Theta_m \cdot \frac{(a^+ - a^-)((a^+ - a^-) \cdot \delta)^{2s}}{(a^+ + a^-)((a^+ - a^-) \cdot \delta)^{2s} + 1} \tag{2}$$

where δ and s are inversely proportional values that determine the individualized trust increment or decrement based on satisfaction and dissatisfaction with interactions. Based on the general frequency of the sparse interactions in an opportunistic network [7] and the empirical evidence from our experiments, this δ should be in the range of (0, 0.05], and it is mapped to the individualized privacy sensitivity of the context type m, ε_m, ($\delta \in (0, 0.05] \mapsto \varepsilon_m$). The value Θ_m weights the value for context m as shared by j based on the cost of retrieving the particular context value. Intuitively, this gives more "credit" to users or devices that share context that is more expensive to acquire in the first place.

As the topology of a pervasive computing network can be sparse and frequently changing, there is a considerable chance that no previous interaction will have occurred between two users regarding the context type m. It is also possible that the resulting trust level is a value that will likely lead to an undesired sharing option later in equation 4. To bootstrap sharing in such circumstances, Magpie considers a *context-similarity* parameter $\Re(m, n)$. This metric provides a measure of similarity between m and n, a second type of context; such a metric could be based on the comparison of the distinct keywords used to describe them [19]. As an example, school information and field-of-study could be considered similar because they both relate to one's educational background. Thus, Equation 2 can be refined as:

$$V'_{a^i_{j,m,k}} = \Re(m, n) \cdot \Theta_n \cdot \frac{(a^+ - a^-)((a^+ - a^-) \cdot \delta)^{2s}}{(a^+ + a^-)((a^+ - a^-) \cdot \delta)^{2s} + 1} \tag{3}$$

where a^+ and a^- are the interaction satisfactions with user u_j regarding to context type n. Of course, a given context type m may be "similar" to more than one other context type; we capture this in Magpie through multiple applications of Equation 3 for different values of n.

At last, we provide support for *situational trust* as a short term trust boost [32, 33]. This short-term situational trust is applied to increase the trustworthiness between a group of users by some adaptive percentage β_k when they

[3] In the equation, $a^+_{j,m}$ and $a^-_{j,m}$ haven been replaced with a^+ and a^- for simplicity.

are perceived to be in some special shared situation. For example two users with a mutual friend may both attend a party hosted by this friend where their joint attendance at the party can bootstrap sharing some context types when the interacting parties are in the same situation.

Considering this last piece of trust determination, Algorithm 1[4] shows the complete procedure of calculating the trust value of a context requester.

Algorithm 1: Instantaneous Trust Calculating Procedure

input : j, peer making request; m, context type; k, current time step
output: $\tau^*_{j,m}$, instantaneous trust value for peer j

1 initialization: $\tau^*_{j,m}$, $\tau_{max} \leftarrow 0$;
2 $V_{a_{j,m,k}} = \Theta_m \cdot \frac{(a^+_{j,m}-a^-_{j,m})((a^+_{j,m}-a^-_{j,m})\cdot\delta)^{2s}}{(a^+_{j,m}+a^-_{j,m})((a^+_{j,m}-a^-_{j,m})\cdot\delta)^{2s}+1}$;
3 **if** $V_{a_{j,m,k}} > 0$ **then**
4 $\quad \tau_{j,m,k-1} + \omega \cdot V_{a_{j,m,k}}(1 - \tau_{j,m,k-1})$;
5 **else**
6 $\quad \tau_{j,m,k-1}(1 - \omega + \omega \cdot V_{a_{j,m,k}})$;
7 **end**
8 $\tau_{max} \leftarrow \tau_{j,m,k}$;
9 **if** *sharing option* $o_{i,m} < 2$ **then**
10 $\quad$ **foreach** c_n *where* $\Re(m,n) > thld$ **do**
11 $\quad\quad V_{a_{j,n,k}} = \Re(m,n) \cdot \Theta_n \cdot \frac{(a^+_{j,n}-a^-_{j,n})((a^+_{j,n}-a^-_{j,n})\cdot\delta)^{2s}}{(a^+_{j,n}+a^-_{j,n})((a^+_{j,n}-a^-_{j,n})\cdot\delta)^{2s}+1}$;
12 $\quad\quad \tau' \leftarrow \tau_{j,n,k-1}(1 - \omega + \omega \cdot V_{a_{j,n,k}})$;
13 $\quad\quad$ **if** $\tau' > \tau_{max}$ **then**
14 $\quad\quad\quad \tau_{max} \leftarrow \tau'$
15 $\quad\quad$ **end**
16 $\quad$ **end**
17 **end**
18 **if** *Situation k perceived* **then**
19 $\quad \tau^*_{j,m} \leftarrow (1 + \beta_k)\tau_{max}$
20 **else**
21 $\quad \tau^*_{j,m} \leftarrow \tau_{max}$
22 **end**
23 **return** $\tau^*_{j,m}$

Line 2 of Algorithm 1 applies Equation 2 to compute the aggregate prior satisfaction and dissatisfaction of user i sharing context type m with peer j. Based on whether this prior is positive, i computes a preliminary trust value for j (specific to context type m) based on Equation 1 (lines 3-8). If this value is likely lead to an undesired sharing option later in Equation 4 (line 9), the algorithm successively applies Equation 3 for each context type n that is "similar" to m (with a similarity value above some specified threshold, *thld*). If this results in a larger trust value than the calculation based on the experiences just with

[4] We omit the i as super script for variables; each step in Algorithm 1 shows the perspective of the user i who is responding to a request from peer user j.

context type m, Algorithm 1 updates the working trust value for peer j. Finally, Algorithm 1 checks whether i and j are in any special shared situation that would boost the trust level that i has computed for j (lines 18-22).

The instantaneous trust value $\tau^*_{j,m}$ returned from the last step (line 16 to 20) is different from the stored trust value that user i maintains for peer j. This returned trust value may indirectly impact the stored trust value in the long term, since it will be used to support interactions, and the user's satisfaction (or dissatisfaction) may cause an update to $\tau^i_{j,m,k}$ for some later value of k.

3.2 Privacy Preserving Sharing of Context

Above, we described how Magpie expressively determines a trust value for a collaborating peer requesting access to a potentially sensitive piece of context information. In this section, we describe how Magpie uses this value to determine what strategy to use when sharing the particular type of context information with the given requester. Magpie's options range from the best possible sharing, which shares the complete raw context information, to sharing no information at all, with Magpie's novel privacy-preserving sharing mechanisms providing a middle ground. The latter can share an obfuscated version of context that considers both the device's context-dependent privacy sensitivity and the trust level that the device has in the particular requesting peer.

Intuitively, the only way to completely avoid any risk of privacy breach is to reject every request for context sharing. But this negates any possible advantage that may come from sharing context information, including learning more broadly about one's surroundings or distributing the costs associated with context sensing. To balance the potential for leaking private information with the benefit to be garnered by sharing context information requires a rational calculation to keep the risk within acceptable limits. Magpie achieves this balance by exposing options that disclose blurred versions of context information when the recipient is not trusted enough to receive the raw data.

Consider a simple example in which a lunchtime line forms at a food truck outside a large office building. Someone still inside the building wonders how long the line currently is in an effort to determine whether it is a good time to get lunch. The device of someone in line could respond to this request in a variety of ways. A naïve user might choose benevolence and be perfectly willing to share information about the line. However, even sharing just this simple piece of information might leak very sensitive private information. For instance, if the user is in line, he is obviously not in his office. This could be sensitive if his coworkers or supervisers expect that he is in a meeting right now. On the other hand, someone else who also knows where his office is might know that now is a good time to steal some of his candy stash. A more cautious user may then want to carefully consider whether the risk of sharing the context information is worth the benefit. There are a few things to consider before participating in the potentially risky behavior. First is the question of *who* is making the request. In real life, if the requester is a buddy of the person in line, they may be completely trustworthy. In the digital world of Magpie, we assume that if

the requester is another user who has proven to be a reliable information source for similar types of information in the past, then a user is may be more willing to reciprocate and provide the requested context information. This is a basic overview of how MAGPIE's adaptive trust management component informs the context sharing actions that users' devices take. As described previously, this process also depends on the particular *type* of context being requested and how sensitive the owner of that data is to sharing it. MAGPIE introduces a *privacy sensitivity* factor to capture this notion.

These first two aspects (i.e., the identity of the requester and the type of context information requested) relate only to the request for the context information. Determining *what* and *how* to share also depends on how well MAGPIE can obfuscate the context information that is shared. In MAGPIE, we achieve obfuscation by adding noise to context information, which can be better achieved when a device has similar context values from other users into which it can *blur* the individual data. In MAGPIE, all such noise additions are computed entirely on the user's personal device using only context information the device has collected or received through other device-to-device interactions. Such an approach is inspired by *differential privacy* and enables MAGPIE to share a blurred version of data with the requester only if the system has enough data to blend the raw data in and make its individual presence appear irrelevant. A similar approach has been used to solve the problem of indirect inference [34], where a composition of pieces of context information that have individually low sensitivity but, when associated with one another could jeopardize a user's privacy. By demanding strict trust in context recipients and offering somewhat inaccurate values, MAGPIE makes it harder to infer such knowledge.

MAGPIE's process for determining what context information to share and how to share it starts with the reception of a request from a peer. Consider the situation when the local MAGPIE system has received a request $r_{j,m}$ from user u_j asking about context type m. Using the algorithm in the previous section, assume that the trust management component determined an instantaneous trust value for this request to be $\tau^*_{j,m}$.

Given a privacy sensitivity for the context type m of ε_m, MAGPIE compares the inner product of $\tau^*_{j,m}$ to ε_m to determine the sharing option:

$$o_{i,m} = \begin{cases} 2 & \text{if } \langle \tau_{i,m}, 1-\varepsilon_m \rangle \geq \theta, \\ 1 & \text{if } \langle \tau_{i,m}, 1-\varepsilon_m \rangle \geq \eta, \\ 0 & \text{if } \langle \tau_{i,m}, 1-\varepsilon_m \rangle < \eta. \end{cases} \tag{4}$$

where θ is is the threshold for being considered as trustworthy as possible for the context type m and η is the threshold for accepting the request; $\theta, \eta \in (0,1)$, and $\theta \geq \eta$. In Equation 4, option codes 1 and 2 indicate that the system will try to accept the sharing request, while code 0 indicates that the request will be discarded. In option 2, the requester exceeds θ, and MAGPIE will simply share the raw context data with the requester. For option 1, meeting or exceeding the threshold η indicates that the requester can be trusted with an obfuscated form of the data, where the level of obfuscation will be further based on the magnitude

of the trust in user j for context type m. For the purposes of this paper, Magpie uses a straightforward approach for both options 1 and 2. For option 2, Magpie simply shares the values generated by the context sensors directly. For option 1, Magpie shares some locally generated statistics, which include aggregating information from other nearby users and adding randomly generated noise.

Magpie's approach builds a trust development ladder, which is important in preventing the overall performance of the Magpie (distributed) system from degrading because devices do not learn to trust one another. Without support from third party relationship sources like social networks [35, 36] (which we aim to avoid), this trust development ladder is essential. That is, an essential component of Magpie is the fact that users can learn to trust each other in *semi-trust* situations as long as the risk can be kept within acceptable limits.

Next we show the basic algorithm that Magpie uses to generate obfuscated context based on aggregating the user's local information with others' information for the same context type and adding random noise. In the end, as we will show, the amount of obfuscation is dependent on the trust value generated for the particular requester and particular context type. Given the sensing neighborhood at the time, let N be the number of recently connected participants for which a reasonably up to date value of context type m is known by the local device; we assume that these peers have identifiers $1 \dots n$. Let c^m be the device's value for context type m and $C'^m = (c_1^m, c_2^m, \dots, c_n^m)$ be the vector of values of context type m for the N peers. Let $C^m = C'^m \cup c$ represent an aggregate of the local context value with the values of the neighboring nodes. In [11] the authors emphasized that knowledge of the exact community distribution (which they refer to as $f_k^e(x)$) is unrealistic because it requires an infinite population. We use a similar notation $f_k^m(x)$ to represent the approximate neighborhood distribution of the local knowledge of context m with limited population at the time instance k. That is, $f_k^m(x)$ is a *statistic* that is *representative* of C^m. To ensure obfuscation commensurate with the required instantaneous trust level for peer j, we further obfuscate $f_k^m(x)$ as shown in Algorithm 2. Our goal here is to perturb the aggregation to achieve context-dependent privacy protection and then randomly select a context value to share given a range whose size is determined by the trust value $\tau_{j,m}^*$, while ensuring the noise being added is controlled by the privacy sensitivity ε_m, which is particular to the context type m.

Algorithm 2 computes the distribution of local context aggregation $f_k^m(x|\mu, \sigma)$ in its initialization stage, where μ and σ are the mean and standard deviation, as usual. For example, a continuous context *temperature* (shown in Fig. 2, where c^m is the self-perceived context) results in the $f_k^m(x)$ shown in Fig. 3. At line 2 Algorithm 2 first determines how many pieces of noisy context (n_p), based on the product of a *perturbing factor* $\lambda \in (0, 2]$ and the cardinality of local aggregation $|C^m|$, should be mixed into the perturbed distribution. In the next step (lines 3-8), n_p pieces of *white Gaussian noised* contexts will be *independently* generated and added into the perturbed set. In line 9 the algorithm calculates the new statistic of the blurry distribution $f_k^m(y)$ before selecting a random variable from the perturbed distribution within the range of $2(1 - \tau_{i,m})$ in line 10.

Algorithm 2: Obfuscating Procedure

input : C^m, set of context values for type m;
$\tau^*_{j,m}$, instantaneous trust value for peer j and context type m;
ε_m, privacy sensitivity for context type m
output: c^m_o, obfuscated context value of type m

1 initialization: $f^m_k(x|\mu,\sigma) \sim C^m$;
2 $n_p = \lambda|C^m|$;
3 **while** $n_p \neq 0$ **do**
4 $\quad \rho \leftarrow E^{n_p}_W(0,1)$;
5 $\quad c_g = \mu + (1+\varepsilon_m)\sigma\sqrt{2}erf^{-1}_{p_n}(2\rho-1)$ $C^m \leftarrow C^m \cup c_g$;
6 $\quad n_p = n_p - 1$;
7 **end**
8 $f^m_k(y|\mu',\sigma') \sim C^m$; // *perturbed pdf*
9 $c_o = \mu' + \sigma'\sqrt{2}erf^{-1}_y(2E_W(0,1-\tau_{i,m})-1)$;
10 **return** c_o

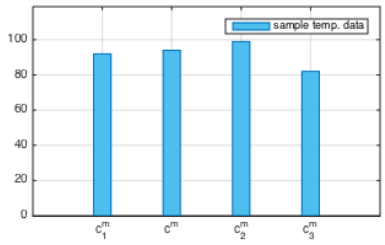

Fig. 2: Local Contexts

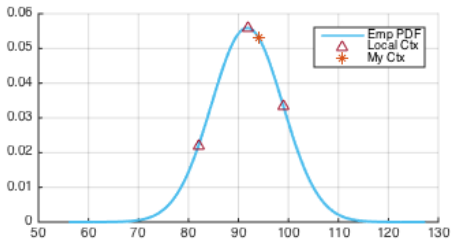

Fig. 3: Empirical Distribution

The loop in lines 3-8 adds n_p pieces of noisy data into the aggregation. Within this perturbed aggregation, the scale of the noise is calibrated to the device's privacy sensitivity for context type m. The error function used in line 5 is from the standard Gaussian statistical noise model except the standard deviation is stretched to $(1+\varepsilon_m)$:

$$P_N(n) = \frac{1}{\sigma'\sqrt{2\pi}} e^{-\frac{(n-\mu)^2}{2\sigma'^2}}, \; where \; \sigma' = (1+\varepsilon_m)\sigma \tag{5}$$

Note that in the process of generating noise, we use the Weibull distributed random numbers [37] ($E_{Weibull} \in (0,1)$; however using other transformation methods should work as well. We also tried the Laplacian noise with $b = \Delta f/\varepsilon$ to determine which perturbation suits our purpose better (Fig. 4). The result complies with the findings in [38] in the sense that the level of noise generated by using the Laplacian model may be so large as to make responses meaningless for many queries for *small data sets* such as a set of evanescent context information; this is why we evaluate MAGPIE using the Gaussian noise model.

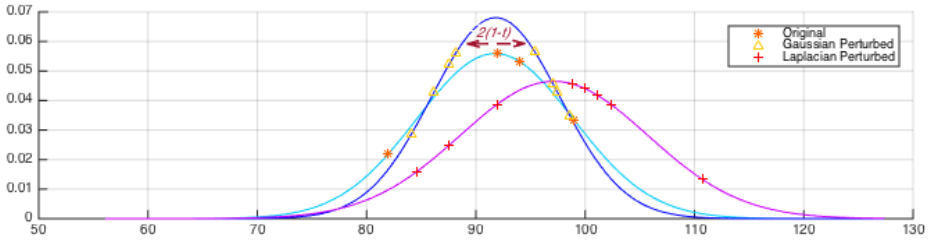

Fig. 4: Perturbed Contexts

4 Experimental Evaluation

To evaluate our proposed approach, we implemented a pervasive context-sharing application as an application protocol in the Opportunistic Network Environment ONE simulator [13]. Each of the Delay-Tolerant Networking (DTN) hosts in the simulation simulates a mobile computing device with embedded sensors, Magpie's adaptive trust evaluation module and privacy sensitivity manager, and an application that periodically *consumes* context information for its own task. When the application's context need cannot be satisfied locally (e.g., because the local host does not have the required sensor) the application generates a context request that it sends to the locally running Magpie system, which disseminates the request to any connected Magpie devices.

Our contributions are two-fold: (1) Magpie facilitates participation in context sharing activities by implementing an adaptive trust scheme; and (2) Magpie protects a context provider's privacy by adding controllable noise into the context being shared according to provider's privacy sensitivity policy and the level of trust between provider and the peer initiating the request. We performed two sets of experiments to evaluate these two contributions.

In our first experiments, we compare the sharing participation of four different schemes: (a) traditional *all-or-nothing* sharing based on a static trust policy; (b) traditional *all-or-nothing* sharing with privacy consideration based on a static trust policy; (c) traditional *all-or-nothing* sharing with Magpie's dynamic trust establishing mechanism; and (d) the full Magpie approach, with both privacy preserving sharing and dynamic trust establishment. To capture the performance in real pervasive computing environments, we conducted the experiments under two settings that entail heterogeneous connectivity protocols, mobility models, and transmit ranges. Table 1 gives the detailed simulation settings.

We ran two different situations: one with 30 nodes and one with 60 nodes. In each, the set of nodes was divided into six equally sized groups as indicated in the table. Nodes were allowed to communicate with other nodes regardless of group. In the table, BT refers to the BlueTooth connection protocol, WiFi refers to standard WiFi links, and highspeed indicates a high-speed and long range wireless interface. The mobility models listed are all built into the ONE simulator, and their names are relatively self-descriptive. The world size parameter in

ONE was set to the same size in both settings ($4500m \times 3400m$), resulting in a *denser* network in the second (i.e., 60 node) case.

	Protocols	**Mobility**	**TX range (m)**	**Speed (m/s)**	**Description**
Group 1	BT	roads	10	(0.5, 1.5)	slow pedestrian
Group 2	BT	pedestrian-path	10	(2.7, 13.9)	car
Group 3	BT & WiFi	tram4	20	(0.5, 1.5)	pedestrian
Group 4	BT & highspeed	mainroads	500	(7, 10)	super connectivity
Group 5	BT	tram10	10	(7, 10)	commuter
Group 6	BT	shops	10	(6, 12)	shop runner

Table 1: Simulation Settings

We first demonstrate the success of MAGPIE in facilitating the sharing of context information among peer devices. We recorded the sharing interactions of the experiments under the four schemes described above to compare how different aspects of MAGPIE affect the community participation in the sharing activity. During the experiment, we simulated five types of context information including three that are continuous measures of ambient context (temperature, light intensity, and noise level), one that is categorical data (power switch) and one that is discrete data (office floor). We run the experiments for 20,000 seconds to ensure that the schemes with trust establishing mechanisms run for a period of time after reaching their stable stages.

In Fig. 5, we show the sum of the number of completed sharing interactions in an experiment lasting 20,000 seconds. There is a noticeable increase (approximately 4×) when MAGPIE's dynamic trust is used (schemes c and d). This suggests that our dynamic trust establishing mechanism explores significantly many more sharing interactions for upper-layer context-aware applications. We can also see that the schemes that employ MAGPIE's privacy sensitivity metrics have slightly lower participation than their counterparts. This indicates that MAGPIE is succeeding in reducing the sharing for privacy preservation by making the decision of selecting the *best possible sharing* strategy context-dependently harder. Finally, it can also be seen that context sharing becomes approximately 10% more frequent in the more densely connected community, which hints at situations in which MAGPIE will be particularly useful.

We next plot the evolution of trust values during the above experiments to understand how the increase in interactivity occurs. We measured the mean trust levels of the context recipients of the same sharing interactions recorded by a single experiment in 10 second intervals. The result is shown in Fig. 6. As this graph shows, the trust level in schemes a and b stays constant throughout the experiment as expected (they both use a static trust model). In schemes c and d, the trust levels oscillate at the beginning and then gradually rise until relatively stabilizing. This observed trends indicate that MAGPIE's privacy preserving sharing helps pervasive devices to become familiar with their surroundings and to establish meaningful trust relationships; this matches our daily social experience:

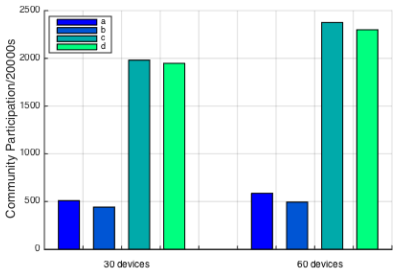

Fig. 5: Sharing Activity Participation

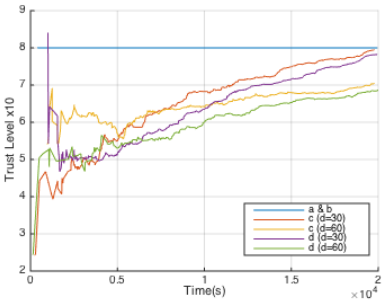

Fig. 6: Trust Establishing Process

we need to be a little extroverted when we arrive in a new place in order to know those who can we get along with and those with whom we cannot.

Magpie's primary goal is to balance an individual's privacy protection against the community's context availability. In our second set of experiments, we take a joint view of a two day long simulation with 60 devices in a larger area ($6000m \times 4500m$). We recorded changes in trust levels, sharing interactions, and quality of shared contexts (as measured by the empirical error [10]) for three context types (with privacy sensitivity (i.e., ε) selected from among $\{0.4, 0.6, 0.8\}$) to investigate how these settings affect each other from an application's view.

Fig. 7 shows the results. The x-axis of all three plots show the elapsed time of the simulation. The middle plot shows the sum of the number of interactions that happened for each type of context in the immediately preceding 600s. The context for which the provider has a low privacy sensitivity (red in Fig. 7) is shared more frequently than medium (green) or high (blue). They have been shared 7.0486, 6.8625, and 5.4722 times per interval on average, respectively. By comparing to the trust level graph in the top of Fig. 7, we can explain this difference as it is apparent that the context with high privacy sensitivity requires a higher level of trust for the provider to participate in this risky behavior.

If we take a closer look at the corresponding trends in the context quality graph (at the bottom of Fig. 7), the least shared type of context (blue in the figure) also results in the the highest percentage of error when it is shared. This is because Magpie shares the obfuscated version of this context in lieu of sharing the raw data, and the privacy sensitivity requires a higher degree of obfuscation than for the other two context types. Note also that the error percentage for all three context types declines over time; this is a result of the gradually increasing trust levels, which result in higher quality sharing as the participants get to know one another better.[5]

[5] Code and full results at: `https://github.com/liuchg/OneSim_PCSharing.git`

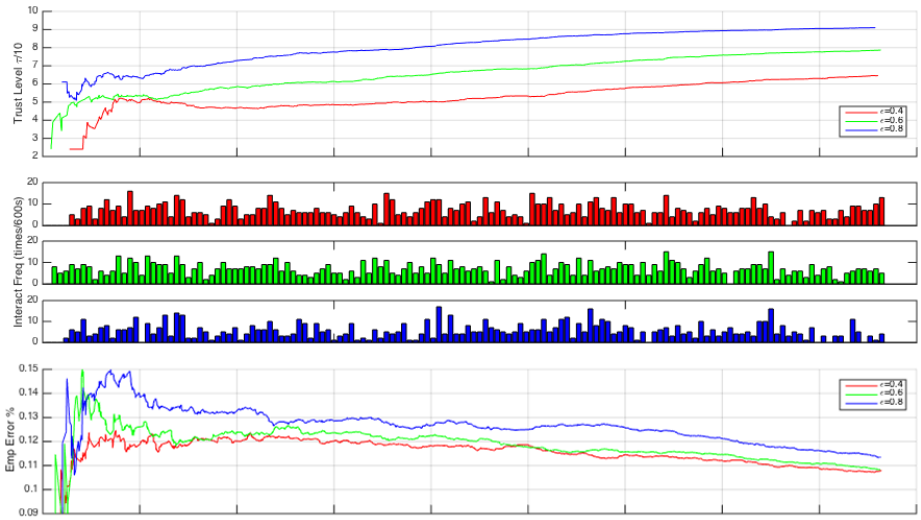

Fig. 7: Joint results from Experiment 2

5 Conclusions and Future Research

Through collaboration, mobile and pervasive computing devices can enjoy unprecedented context availability and help users to exploit the nearby environment. However, sharing context information sensed by a user's personal device poses threats to the user's privacy and must be controlled. We introduced Magpie which, by dynamically evaluating the *risk* of disclosing potentially private data based on the level of trust between the participants and the individualized context-dependent sensitivity, helps users to select sharing strategies for context. In Magpie we assumed trustworthiness to be reciprocal relationship. Future work will explore additional factors to determining the trustworthiness of a collaborating peer, including relaxing this assumption. In our initial work with Magpie, we have demonstrated that there are context types amenable to our simple data perturbation mechanisms. This may not be true for all types of context information; future work will look at specialized ways to add noise to common types of context data to increase the applicability of Magpie. Currently, Magpie responds to each context request individually; it is possible that multiple neighboring devices may request the same or similar information from a user. Optimizations to Magpie's behavior could save some processing overhead by using results of previous computations.

In this paper, we built a prototype of our current vision of Magpie. Given this prototype, we performed a series of application-oriented experiments performed on the ONE simulator. Even without the enhancement discussed above, this evaluation validated that Magpie can effectively facilitate context sharing activities by implementing an adaptive trust scheme and can protect a context provider's privacy by adding controllable noise into the context. We expect that

future work will enhance Magpie's capabilities and extend the types of context to which it is applicable.

References

1. K. Shilton. Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. *Comm. of the ACM*, 52(11):48–53, 2009.
2. S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Trans. on Sensor Networks*, 6(1):6, 2009.
3. D. Mendez, A. J. Perez, N. Labrador, J. J. Marron, et al. P-sense: A participatory sensing system for air pollution monitoring and control. In *Percom Workshops*, pages 344–347, 2011.
4. E. Bales, N. Nikzad, N. Quick, C. Ziftci, K. Patrick, and W. Griswold. Citisense: Mobile air quality sensing for individuals and communities design and deployment of the citisense mobile air-quality system. In *Proc. of PervasiveHealth*, 2012.
5. E. Grim, C.-L. Fok, and C. Julien. Grapevine: Efficient situational awareness in pervasive computing environments. In *Proc. of Percom Workshops*, 2012.
6. M. Srivastava, T. Abdelzaher, and B. Szymanski. Human-centric sensing. *Philosophical Trans. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 370(1958):176–197, 2012.
7. F. Almenarez, A. Marin, D. Díaz, and J. Sanchez. Developing a model for trust management in pervasive devices. In *Proc. of Percom Workshops*, 2006.
8. X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher. Artsense: Anonymous reputation and trust in participatory sensing. In *Proc. of INFOCOM*, 2013.
9. L. Xiong and L. Liu. Building trust in decentralized peer-to-peer electronic communities. In *Proc. of ICECR-5*, 2002.
10. E. Shi, T.-H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Proc. of NDSS*, 2011.
11. R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher. Poolview: stream privacy for grassroots participatory sensing. In *Proc. of SenSys*, pages 281–294, 2008.
12. C. Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*, pages 338–340. 2011.
13. A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Pro. of SimuTOOLS*, page 55, 2009.
14. D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.
15. L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Comm. Mag.*, 44(11):134–141, 2006.
16. N. Luhmann. Familiarity,n.confidence, trust: Problems and alternatives. *Trust: Making and breaking cooperative relations*, 6:94–107, 2000.
17. H. Li and M. Singhal. Trust management in distributed systems. *IEEE Computer*, 40(2):45–53, 2007.
18. S. S. Babu, A. Raha, and M. K. Naskar. Trust evaluation based on nodes characteristics and neighbouring nodes recommendations for WSN. *Wireless Sensor Network*, 2014, 2014.
19. M. G. Uddin, M. Zulkernine, and S. I. Ahamed. Cat: a context-aware trust model for open and dynamic systems. In *Proc. of SAC*, pages 2024–2029, 2008.

20. A. A. Selcuk, E. Uzun, and M. R. Pariente. A reputation-based trust management system for p2p networks. In *Proc. of CCGrid*, pages 251–258, 2004.
21. L. Sweeney. k-anonymity: A model for protecting privacy. *Int'l. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
22. I. Bilogrevic, J. Freudiger, E. De Cristofaro, and E. Uzun. Whats the gist? privacy-preserving aggregation of user profiles. In *Proc. of ESORICS*, pages 128–145. 2014.
23. A. Reinhardt, F. Englert, and D. Christin. Averting the privacy risks of smart metering by local data preprocessing. *Pervasive and Mobile Comp.*, 16:171–183, 2015.
24. G. Pallapa, S. K. Das, M. Di Francesco, and T. Aura. Adaptive and context-aware privacy preservation exploiting user interactions in smart environments. *Pervasive and Mobile Computing*, 12:232–243, 2014.
25. U. Hengartner and P. Steenkiste. Avoiding privacy violations caused by context-sensitive services. *Pervasive and Mobile Computing*, 2(4):427–452, 2006.
26. A. Tomasic, J. Zimmerman, A. Steinfeld, and Y. Huang. Motivating contribution in a participatory sensing system via quid-pro-quo. In *Proc. of CSCW*, 2014.
27. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *Proc. of SenSys*, pages 125–138, 2006.
28. R. Shokri, G. Theodorakopoulos, P Papadimitratos, E. Kazemi, and J. Hubaux. Hiding in the mobile crowd: Locationprivacy through collaboration. *DSC, IEEE Trans on*, 11(3):266–279, 2014.
29. Y. Liu, A. Rahmati, Y. Huang, H. Jang, L. Zhong, Y. Zhang, and S. Zhang. xshare: supporting impromptu sharing of mobile phones. In *Proc. of MobiSys*, 2009.
30. N Golrezaei, A Molisch, A G Dimakis, and G Caire. Femtocaching and device-to-device collaboration. *IEEE Comm. Mag.*, 51(4):142–149, 2013.
31. A. Oulasvirta. Finding meaningful uses for context-aware technologies: the humanistic research strategy. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 247–254, 2004.
32. M. Stephen. Formalising trust as a computational concept. *PhD dissertation. University of Stirling, Scotland*, 1994.
33. C. Duma, N. Shahmehri, and G. Caronni. Dynamic trust metrics for peer-to-peer systems. In *Proc. of DESA*, pages 776–781, 2005.
34. X. Jiang, J. Landay, et al. Modeling privacy control in context-aware systems. *IEEE Pervasive Computing*, 1(3):59–63, 2002.
35. Y. Lu, Z. Wang, Y.-T. Yu, R. Fan, and M. Gerla. Social network based security scheme in mobile information-centric network. In *Proc. of MED-HOC-NET*, 2013.
36. I. Parris, G. Bigwood, and T. Henderson. Privacy-enhanced social network routing in opportunistic networks. In *Proc. of Percom Workshops*, pages 624–629, 2010.
37. Yu.K. BelyaevE.V. Chepurin (originator). Weibull distribution. `http://www.encyclopediaofmath.org/index.php?title=Weibull_distribution&oldid=18906`.
38. R. Sarathy and K. Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Trans. on Data Privacy*, 4(1):1–17, 2011.