# AMBIENCE: A Context-Centric Online Social Network

Martin Werner
Mobile and Distributed Systems Group
Ludwig-Maximilians-University Munich
Munich, Germany
Email: martin.werner@ifi.lmu.de

Florian Dorfmeister
Mobile and Distributed Systems Group
Ludwig-Maximilians-University Munich
Munich, Germany
Email: florian.dorfmeister@ifi.lmu.de

Mirco Schönfeld
Mobile and Distributed Systems Group
Ludwig-Maximilians-University Munich
Munich, Germany
Email: mirco.schoenfeld@ifi.lmu.de

***Abstract*—Context-awareness has been discussed for a long time and is a primary driver of personalization, which is crucial to mobile computing. Additionally, online social networks usually contain enough highly private information for personalization, but this information has been seldom used in modifying the social graph structure. With this paper, we define the novel paradigm of context-centric online social networking in which the social graph structure is derived from context-information alone. We show how a context-centric network based on Wi-Fi location information as well as geohash-based GPS location can be deployed using a specific metadata construction. We support our definitions with a location-based service relying on the metadata model and illustrating spatial and temporal awareness.**

## I. Introduction

Context-aware computing is a key step towards raising computing experience to a new level. Context information provides an information source for adaption of applications and networks to the actual and current user needs and enables better service experience as the service fits the need of the users and complex interactions between different applications and users become feasible. This is due to the fact, that the behavior of others is an integral part of the situation of an application and thus provides context information for the application. Hence, applications can be made aware of each other and this awareness can be used to provide better service quality [15].

Context-awareness with respect to a person, often called personalization, has been well-researched in the past and is typically based on collecting information about the physical surroundings of the person, most often the location [1]. Due to the rise of communication in social networks, it has become possible to collect additional information about the social ambience connected to a profile in an online social network representing a person. This type of context information can additionally be used to provide better personalization. However, this information also includes a lot of knowledge about the information flow inside the network and can be exploited to provide higher quality-of-service from a network perspective. This approach has been widely used in the area of sensor networks with the battery level and energy consumption as the primary context information [3], [9].

In the last decades, novel communication patterns have been observed inside the Internet, especially due to online social networks. These novel communication patterns are not well supported by classical address-based Internet routing and gave rise to the paradigm of information-centric networking [2]. In information-centric networking, the network is organized around the information dissemination need rather than around sources and destinations providing better support for these novel traffic patterns.

With this paper, we intend to bring these isolated domains together and provide communication adaption from user context information for context-aware online social networks. Therefore, we propose the novel and tighter paradigm of context-centric online social networks, in which communication is not based on virtual friendships anymore, but solely on context similarity. These context-centric online social networks provide means to improve privacy and simplify targeted communication in larger and dynamic social groups defined from context and, hence, provide even better communication experience than in online social networks themselves. One can also see a context-centric online social network as an online social network in which friendship is based on context similarity alone.

With this paper, we define the paradigm of **context-centric online social networks** and design and implement a context-centric online social network using a specific meta data construction which – in contrast to typical location-based online social networks like Sindbad [13] – is not bound to any specific type of context information.

Section II provides a short overview of context-awareness and gives definitions of semi-constant and quasi-random context. Further, information-centric networking is introduced. Finally, the Bloom filter construction is explained and the central observation of how Bloom filters bitsets can be compared using a variant of the Jaccard distance between sets gets introduced. In Section III, the novel paradigm of context-centric online social networks is defined. In Section IV, a metadata format for context-centric online social networks is constructed and illustrated in several application scenarios. In Section V, our prototypical implementation of the ideas presented in this paper is described. Finally, Section VI concludes the paper.

## II. Background

With this paper, we introduce a novel paradigm of *context-centric online social networks*. These are context-aware, information-centric online social networks in which information dissemination is tightly bound to context similarity. With

978-1-4799-8923-2/15/$31.00 

this section, we recollect the needed background and related work from the various fields which we intend to bring together into a novel type of social network. Therefore, we describe the area of context-aware computing, the paradigm of information-centric networks, the central data structure of a Bloom filter, and the paradigm of online social networks. In the section about the Bloom filter, we explicitly state the observation that the Bloom filter allows for blind calculation of set similarity, which – to the best of our knowledge – has not been stated in this way before. However, the needed ingredients have been given in other work; therefore, we give this *central observation as part of the background* section [16].

### A. Context Awareness

Though there are a lot of previous definitions of context information and context awareness, the following definition by Dey and Abowd has been widely accepted for the general scope and simplicity of their definition:

*"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application including the user and application themselves." [1]*

This definition leaves several central elements undefined and thus incorporates common sense about what actually is the concrete meaning of terms such as information and situation. This can be explained from their application-centric perspective. They state that the main power of their definition is that this definition makes it easier for an application developer to enumerate possible application context in concrete cases.

For this work and with a perspective towards context-centric networking in mind, we want to extend and concretize the definition to meet current and future developments as follows:

**Definition 1** (Context)
*Context is any information that can be used to infer aspects of the surroundings of an entity in a way, in which some applications might have interest. Surroundings include all information that could possibly impact the behavior of the entity.*

The concept of context-awareness has first been mentioned by Schilit et al. [14] as a paradigm for research in ubiquitous and pervasive computing. It is based on the general idea of computing systems being able to sense both their users' as well as their own context information, perform inference mechanisms thereon and adapt their behavior accordingly. Since then, a plethora of different usage scenarios for context-aware applications have been proposed. However, the most common type of context-awareness is given by location-based services (LBS), which – due to their vast number of different application scenarios presented – can so far be regarded as the most prominent example of context-aware applications.

Context information itself is commonly divided into context atoms and higher-level context. Although it is difficult to impose any strict boundaries between these two categories, context atoms can generally be directly sensed by some kind of hardware sensor (such as, e.g., device acceleration, light intensity or Wi-Fi signal strengths), whereas higher-level context usually results from aggregation of several context atoms or some other kind of inference mechanism performed on the latter. Note that in practical scenarios, the sensor interface is often used to distinguish atomic context and higher context: For example, a GPS receiver consists of several internal "sensors" interacting with each other and the task of obtaining GPS location splits in sensing tasks including time-synchronization, almanach download and actual signal timing, which all could be seen as isolated atomic context. But for applications which do not access these raw measurements, the atomic context is the WGS84 position calculated from these internal measurements.

We define two special classes of context information as they are crucial to different types of adaption: *Semi-constant* context information and *quasi-random* context information.

*Semi-constant context information* is context information with a limited rate of change. Semi-constant context variables can be used to improve energy efficiency by suppressing measurement and communication for context-variables, whose worst-case change does not yield an effect for an application. The most prevalent and well-researched example is location, where measurements for neighborhood tracking can be deferred until there is a chance of two items being near. The location context is assumed to be changing with a maximum speed. All possible locations, where a target can be after some time, are given by the inner area of a circle around the last position. However, there is a lot of semi-constant context information such as humidity, pressure, air pollution for the class of atomic context variables as well as user activity for the class of higher-level context. In addition, a user's profile information can also be regarded as another kind of semi-constant context information, likely to have a very low rate of change. However, this information can be used as a key element for establishing preference based context similarity, allowing for fine-grained detection of communities and sub-communities within groups.

On the contrary, *quasi-random context information* is context information for which it is impossible (or hardly possible) to infer the actual context information without measurements at the same location and situation. Quasi-random context information is best used to derive keys in a distributed communication network protected by encryption. There is a lot of research done on secure key extraction from surroundings (e.g., context) as in [8], [11]. With these techniques it is readily possible to encrypt data for decryption by entities which have been roughly at the same location at the same time. It is worth noting, that such a geo-encryption scheme does not lead to a cryptosystem in which only nearby users are able to decrypt data. It is also possible for a user to publish the location-dependent key allowing all entities to decrypt data that was published with a location-dependent key.

### B. Information-centric Networks

Information-centric networks have evolved in different scenarios of Future Internet. While the Internet is currently organized around end-to-end communication, a lot of services need reliable, scalable and efficient information dissemination. This is often done by constructing peer-to-peer and content delivery networks as overlay networks based on end-to-end communication.

With the deployment of a truly information-centric networking, new naming schemes become mandatory. In classical Internet naming a URI identifies the data as well as the network address (e.g., IP address) from where to obtain a specific data object. This naming scheme has severe limitations, and does not allow for data mobility. If a data object moves on from one server to another, its URI changes. The only possibility is to leave a redirection behind at the old URI. Furthermore, if the number of users exceeds the server capacity, the URI might become unreachable.

An information-centric network is a fundamentally different approach in which three building blocks have to be defined [2]: a definition of information objects and their associated different representations, a naming scheme, which is sufficiently location-independent and provides the needed security mechanisms, and an application programming interface, which makes the operation of information-centric networks transparent to applications.

For information-centric networks to become a valuable key element in Future Internet there are still a lot of challenges including naming, scalability, security, privacy, mobility and object localization [2].

In most information-centric networks, metadata is exchanged among nodes to provide network services. *Metadata* is a representation of the contents of an information object suitable for concisely describing the information object content. Metadata typically includes the size, content type, hash values and other technical details directly calculated from the information objects data block as well as semantically meaningful additional information such as: "This object contains a photo showing Bob". For information-centric network operation, metadata should be smaller than the information objects themselves and sufficient for non-interested peers to take reasonable routing and caching decisions. Metadata should be able to provide additional mechanisms for reducing node complexity, most importantly aggregation, merge and distributed reconciliation. Metadata provides aggregation if it is possible to calculate a summarizing metadata representation out of a set of objects. This summarizing metadata is called aggregated metadata in the sequel. Metadata provides an aggregated merge operation, if it is possible to calculate the aggregated metadata of the union of two sets out of the aggregated metadata of the individual sets. Metadata provides for set reconciliation if there is an algorithm, which efficiently calculates a list of information objects that two nodes do *not* have in common based on the aggregated metadata of the two sets of objects each node have in their cache.

There is a very widely used and highly generic approach for information object metadata using Bloom filters and their numerous variants.

### C. Bloom Filter

A Bloom filter is a probabilistic data structure describing sets. A Bloom filter, in its simplest form, consists of a bit array of some fixed length and is based on calculating a fixed number of hash functions of the information object taking positive integer values smaller than the length of the hash field. These hash values can be used to set the bits, if an information object is added to the filter and for querying the filter, these hash values have to be checked. If one of the associated hash values for a query is not set, the object has certainly not been added to the filter. Bloom filters provide aggregation, merge and set reconciliation [4], [5].

With respect to the constructions in this paper, it is important to know that the fraction of zeros of a Bloom filter (i.e., the number of bits that are zero divided by the overall number of bits) can be used to estimate the size of a filter:

*Given a Bloom filter array $F$ of a filter with $m$ slots and $d$ hash functions, calculate the fraction of zeros $\phi$ of the filter. Then the following equation estimates the number of elements in the filter:*

$$n_F \approx -\frac{\log(\phi)m}{d}$$

Note that this estimator returns a real number and that it might be reasonable to round this number to the next integer in applications.

As it is possible to calculate the union of two Bloom filters of identical configuration by just using a binary OR on their hash fields, the number of elements of the union of two filters $F$ and $G$, each configured with $m$ slots and $d$ hash functions, can be calculated from both filters filters. In this case, for the union of the filters, we calculate

$$n_{F\cup G} \approx -\frac{\log(\phi_{F\cup G})m}{d},$$

where $\phi_{F\cup G}$ is the fraction of zeros of the binary OR of both filters.

These two relations can also be used to estimate the number of elements of the intersection of both filters by using the set identity

$$|A \cup B| = |A| + |B| - |A \cap B|$$

resulting in

$$n_{F\cap G} \approx n_F + n_G - n_{F\cup B}$$

Again, this number can be rounded in applications, however, when calculating further, it can be reasonable to work on real values.

These two observations allow for approximating the Jaccard distance of two sets:

**Theorem 1**
*The Bloom filter datastructure allows for the approximate calculation of the* ***Jaccard distance***

$$\delta_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

*for two equally configured filters $F_A$ and $F_B$ using*

$$\delta_J(F_A, F_B) \approx 1 - \frac{n_{A\cap B}}{n_{A\cup B}}$$

We will make use of this observation when constructing our context-centric online social network in Section V.

For the Bloom filter construction, two variants can be of great importance: The counting Bloom filter and the time-decaying Bloom filter. In both cases, the bit fields of the Bloom filter are replaced by small integer counters. For *counting Bloom filters*, each time an element is added to a filter, the associated elements are increased by one. In this way, the number of times a specific element has been inserted to a filter can be approxiamted by an upper bound. So the filter never claims that fewer elements have been inserted than have actually been inserted. However, it can overestimate the number of elements in a filter due to hash collisions. For *time-decaying Bloom filters*, the counters are increased by a fixed integer amount $r$ modelling the number of rounds, that the element shall be contained in the filter. In each round, all cells with positive values are decremented by one. In this way, the value $r$ models the number of rounds (e.g., the duration) that the counters addressed by the various hash elements keep non-zero.

### D. Distributed Online Social Networks

An online social network is an online service, which allows for communication between individuals based on their social network. There have been several different definitions of online social networks mainly differing in the way they relate to definitions of social networks from sociology. A pretty concrete and widely accepted definition of online social networks has been given by Boyd and Ellison in [7]. They use a more restrictive terminus "social network site", which we take synonymous for the broader terminus "online social network" used in this work.

We define social network sites as web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site. This classical definition can be mapped onto a central data structure of social networks, the *social graph* sometimes called friendship graph. The social graph is a graph containing one vertex per profile and an edge between two users, if they want to share data with each other. Beside the social graph, there is another prevalent graph structure used to describe and analyze online social networks: The *interaction graph.* In typical social networks, a lot of social graph edges lie between people who do not really interact. This problem is referred to as *social network pollution* [6].

In online social networks suffering from social network pollution, a better view on the underlying social network is given by online social network interactions. These include active interactions such as posting text, commenting pictures, etc. as well as passive interactions such as browsing a profile, reading postings, etc. The interaction graph is a multigraph in which nodes represent profiles and edges between nodes represent interactions.

Online social networks are often provided using a client-server architecture and distribution to several physical nodes is often only used to provide scalability to a logically central service. In this setting, the central logical entity knows the complete social graph and can provide help with constructing links between profiles in various ways. However, this architecture has some disadvantages for the users, namely vendor lock in, commercial exploitation of their private information, and general privacy risks.

Therefore, research has focused on providing distributed online social networks. A distributed online social network is an online social network in which no central entity exists. The network functionality is provided in a peer-to-peer manner between users. It is not difficult to provide online social network services in a decentralized manner since communication in an online social network is usually traversing along edges of the social graph. However, *permanent profile availability* and *efficient distributed search* are becoming more complex in this setting.

## III. Context-Centric Online Social Networks

Motivated from the idea of providing adaptive networks using the information-centric networking paradigm and due to the fact that adaption has been most successfully provided from context and that online social network communication has a traffic pattern for which context and information-awareness can be advantageous, we bring these fields together with our following definitions:

**Definition 2** (Context-Aware Online Social Network)
*A* ***context-aware online social network*** *is an online social network in which the edges in the social graph are annotated with context variables and in which it is made possible to share information based on context similarity.*

An illustrative example could be given by the publish operation for a photo as follows: "Share this picture with all my friends, who I physically met with confidence higher than 80% in the last two weeks."

Context-aware online social networks can easily be seen as an extension to usual online social networks defining virtual groups and privacy settings based on context information. We want to further extend these notions to a more stringent notion by defining context-centric online social networks as follows:

**Definition 3** (Context-Centric Online Social Network)
*A* ***context-centric online social network*** *is an online social network in which the edges of the social graph are defined from context information and context matching algorithms. An edge between two profiles exists for a fixed information object if and only if the two profiles share the relevant context as defined by the publisher.*

Note that in context-centric online social networks as defined above communication is quite restricted. Only if the publisher and the subscriber fulfill some constraint on context similarity, they communicate the information object. This can have severe impact on the information dissemination performance when the number of users who forward specific information objects is too low. One of the motivations of defining context-centric online social networks in this way, is, that these networks have no social network pollution by definition. Edges are defined from actual and contemporaneous social network interaction and hence no inactive virtual friendships can exist.

## IV. Towards a Context-Centric Online Social Network

In this section, we describe how to use the well-known and widely deployed Bloom filter construction in order to create a working prototype of a context-centric online social network. With this part, we concentrate on the context-centric functionality and employ data structures that provide distributed operations. For simplicity, however, we construct this first context-centric network in a client-server manner. The most important consideration, however, concerns metadata management and is explained in the next section.

### *A. Metadata in Context-Centric Networks*

With this section, we explain how the classical Bloom filter construction can be used to create information metadata with the following features useful for deploying a context-centric online social network. Therefore, the context information is represented in whole as a set of string labels, which are put into a Bloom filter using a secure hash function resulting in:

- *Typeless Representation:* The representation of information is completely decoupled from data types or data object sizes due to using hash functions on binary representations of the data.
- *Blind Subset Operation:* Without getting to know the content that some meta-data object describes, subsets can be detected with a small probability of false decisions.
- *Approximate Jaccard Similarity:* Without knowing the contents described by the meta-data objects, it is possible to approximate the distance of two filters in the sense of the Jaccard metric calculated from the estimated Jaccard index given in Theorem 1.

These three features allow for the following interpretation, when we fix a novel addressing scheme in which the address of a message is defined to be a small set of strings represented as a Bloom filter of previously known configuration:

**Observation 1**
*We can address a single message to a small subset of strings and select messages from a database based on the Jaccard set similarity of these sets of strings or based on the subset relation.*

This addressing scheme will be used and the following three examples illustrate, how this can reflect context and context similarity in a quite natural way: *Spatial awareness inside buildings* can be generated from using MAC addresses of Wi-Fi access points in the surroundings. A message can be bound to a set of MAC-adresses and the two operations of selecting messages that are subsets of the set of access points visible to a mobile device at a specific location and time or selecting messages which have a similar set of access points in their address can realize both: *location-limited publication* of messages and *location-based ranking* of messages based on set similarity. *Geospatial awareness based on coordinates* can be generated using a Z-curve coding similar to the well-known geohash [10]. Using a Z-curve coding of limited length, the message can be bound to a location. Fortunately, a client can build spatial queries using any set of grid cells inside this framework and calculating grid geohashes of neighboring cells is very easy. This application is similar to the GeoCookie construction proposed by Ruppel et. al. [12]. When adding a location, we could also add the neighboring cells for larger query support. *Activity information* can be used by adding strings from a previously chosen set of labels. We can, for example, add the next upcoming activity such as *Playing Football* to the filter. By properly using subset or similarity queries, we can then select messages based on such discrete context values.

These examples show that the framework is completely general and the only limitation is with respect to the type of queries and an elegant representation of continuous variables.

### *B. Group Messages and Epoch-based Privacy*

With our proposal, context situations are subsumed by small sets of strings and represented by Bloom filters. If the hash functions of the Bloom filters are cryptographically secure, we can expect that users can't find out the strings that have actually been added to a filter. However, filter replay attacks can be made. Each user that has once been in a specific context situation is able to calculate the associated filter and make targeted queries.

This creates a privacy problem as an attacker has unlimited time to create a database of realistic filters and use these filters to circumvent the assumed context-centrality of the system and retrieve messages which are intended for users having this context now.

In order to tackle this problem, we propose to use random prefix strings in order to limit the number of people that can actually do a directed query based on context to a specific group:

**Observation 2**
*By choosing a prefix for the hash functions and using cryptographic hash functions, we can model access control on the contents of the data structure without hiding the approximate number of elements for any filter and the expected set similarity for two filters created using the same prefix.*

This can be used to form different types of groups.

- A **group** of users chooses a secret, common prefix. Only those who know this string or are able to brute force this string using known location of some group element can find out the relevant messages for the group. Note, however, that the messages are not protected in any way. Retrieving a message is easy and possible, however, the directed retrieval of messages of a group is hard.
- The platform creates **time epochs** in which all publications have to use a platform-defined, long, random prefix. Then, a time-targeted retrieval of information is only possible for users, who know the epoch.
- A **spatially limited time epoch** can be constructed by ensuring that only users in a specific location at a specific time can retrieve the prefix string for this spatio-temporal epoch.

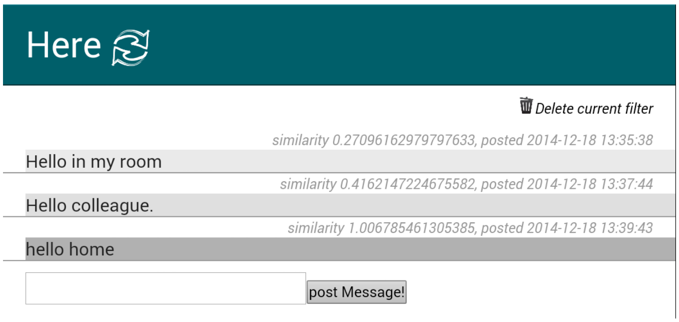


Fig. 1. Screenshot of the prototype

## V. Ambience: A Context-Centric Online Social Network Prototype

We implemented a proof of concept application realizing the context-centric network developed in the previous sections. It is based on a central server, and two different client components: one for publishing messages and one for retrieving the most similar messages from the service.

The server stores and manages a list of pairs of a Bloom filter and a multimedia message (text, image, link, ...). Additionally, the server provides an interface for querying the $k$ most similar Bloom Filters or the $k$ most similar Bloom filters that contain a specific other Bloom filter.

The subscription part of the client regularly maintains a time-decaying Bloom filter and requests a ranked, limited list, ordered by time, attributed with the estimated Jaccard similarity from the server. This is shown to the user of the service in different ways. We especially deployed a coloured sidebar for each post which illustrates the Jaccard similarity of a specific post with the query. In this way, the two orderings time and similarity can be shown at the same time in a relatively intuitive way. For publishing messages, the same Bloom filter is being used in order to publish a post at the "current" contextual situation.

Figure 1 shows a screenshot of the application. The virtual filter "Here" correspondes to the current location and shows the ten most context-similar messages ordered by time. In this case, three messages are shown, which were taken in my room (white background), in my colleagues room (distance 15m), and my home (distance 10 km). You can see that even small distances (15m) are taken into account by the system while the far-away message is considered completely irrelevant. Note that the system also allows me to save a filter and give a name to it. This serves like a channel subscription: by clicking on such a stored filter, I retrieve the most similar messages from a pre-defined time window of the past. In summary, the system worked as expected and revealed that the management of different channels and the tradeoff between relevance, time and number of messages can become complicated.

## VI. Conclusion

With this paper, we present the novel paradigm of context-centric online social networks, a metadata approach for actually constructing context-centric online social networks as information-centric networks and a demonstrative implementation called "Ambience" in which users can share text-messages based on both: context similarity and temporal actuality.

Future work will focus on the security and privacy mechanisms of the system and on a context-centric encryption scheme in order to also protect the message content based on context information. Aside that, we are convinced that the novel domain of context-aware and context-centric networking provides a lot of open and exciting research problems, which should be taken care of by the community.

## References

[1] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[2] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26–36, 2012.

[3] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.

[4] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4):485–509, 2004.

[5] John Byers, Jeffrey Considine, and Michael Mitzenmacher. Fast approximate reconciliation of set differences. Technical report, Boston University Computer Science Department, 2002.

[6] Michael Dürr, Martin Werner, and Marco Maier. Re-socializing online social networks. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 786–791. IEEE, 2010.

[7] Nicole B Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.

[8] Suman Jana, Sriram Nandha Premnath, Mike Clark, Sneha K Kasera, Neal Patwari, and Srikanth V Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 321–332. ACM, 2009.

[9] Christine E Jones, Krishna M Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *wireless networks*, 7(4):343–358, 2001.

[10] Gustavo Niemeyer. public announcement of geohash.org, 2003. http://forums.groundspeak.com/GC/index.php?showtopic=186412.

[11] Sriram Nandha Premnath, Suman Jana, Jessica Croft, Prarthana Lakshmane Gowda, Mike Clark, Sneha Kumar Kasera, Neal Patwari, and Srikanth V Krishnamurthy. Secret key extraction from wireless signal strength in real environments. *Mobile Computing, IEEE Transactions on*, 12(5):917–930, 2013.

[12] Peter Ruppel and Axel Küpper. Geocookie: a space-efficient representation of geographic location sets. *Journal of Information Processing*, 22(3):418–424, 2014.

[13] Mohamed Sarwat, Jie Bao, Ahmed Eldawy, Justin J Levandoski, Amr Magdy, and Mohamed F Mokbel. Sindbad: a location-based social networking system. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 649–652. ACM, 2012.

[14] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[15] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *Workshop Proceedings*, 2004.

[16] S Joshua Swamidass and Pierre Baldi. Mathematical correction for fingerprint similarity measures to improve chemical retrieval. *Journal of chemical information and modeling*, 47(3):952–964, 2007.