Toward a Distributed k-Anonymity Protocol for Location Privacy

Ge Zhong and Urs Hengartner
Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzhong,uhengart}@cs.uwaterloo.ca

ABSTRACT

To benefit from a location-based service, a person must reveal her location to the service. However, knowing the person's location might allow the service to re-identify the person. Location privacy based on k-anonymity addresses this threat by cloaking the person's location such that there are at least $k-1$ other people within the cloaked area. We propose a distributed approach that integrates nicely with existing infrastructures for location-based services, as opposed to previous work. Our approach is based on homomorphic encryption and has several organizations, such as operators of cellphone networks, collaborate to let a user learn whether k-anonymity holds for her area without the organizations learning any additional information. We also outline several challenges that remain to be addressed.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*; E.3 [**Data**]: Data Encryption—*Public key cryptosystems*

General Terms

Algorithms, Design, Security

1. INTRODUCTION

With the advance of location technologies, people can now determine their location in various ways, for instance, with GPS or based on nearby cellphone towers. These technologies have lead to the introduction of location-based services, which allow people to get information relevant to their current location. Location privacy is of utmost concern for such location-based services, since knowing a person's location can reveal information about her activities or her interests.

For location privacy based on k-anonymity, a user's current location is cloaked such that there are at least $k-1$ other users within the cloaked area. A location-based service learns only the cloaked area, which allows the user to remain anonymous within the set of k users. Applying k-anonymity to location cloaking has been studied extensively [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15]. Traditionally, this approach has been implemented with the help of a central trusted server [1, 2, 3, 6, 9, 10, 11, 12, 13, 15]. Here, users register their current location with the trusted server. Whenever a user wants to access a location-based service, she has the trusted server compute a cloaked area that has the k-anonymity property. Then, the trusted server contacts the location-based service on the user's behalf. The drawback of this approach is that the trusted server knows everybody's location. Users must trust it not to leak their location information to unauthorized parties, maybe inadvertently. In short, the trusted server is a single point of failure. More recent research has proposed to get rid of the trusted server and to have (nearby) users jointly compute a cloaked area that has the k-anonymity property [4, 7, 8]. Then, the user (or, for increased privacy, another user on her behalf) contacts the location-based service. The drawback of this approach is that all existing solutions trust users to implement a solution faithfully and not to leak location information learned during the computation.

We propose a solution that requires neither a single trusted server nor trust in all users of the system. Namely, we have multiple servers, each deployed by a different organization (e.g., an operator of a cellphone network) and each knowing the location of only a *subset* of users (e.g., the operator's customers), with the subsets being disjoint. When a user wants to access a location-based service, she cloaks her area and asks each server for the number of people in this area. In a naïve solution, the servers simply give her these numbers, she sums them up and, if the sum is at least k, she accesses the location-based service. However, this approach has the flaw that the user could track people. For example, if the user learns that there is only a single person in an area and nobody in the surrounding areas, the user can follow the path of the person when the person leaves the area and enters one of the surrounding areas. As soon as the person enters an area that is associated with her identity or that is under surveillance by the user, the user can re-identify her. In general, sophisticated data-mining technologies might allow the tracking or re-identification of a person even if there are multiple people in an area.

Our solution avoids this problem with the help of cryptography and ensures that a user cannot learn the number of people in an area reported by a server. The user can learn only whether the sum of these numbers is at least k. Moreover, our approach integrates nicely with existing infrastructures for location-based services, where an operator of a cellphone network knows the current location of *its* customers, but there is no single entity that knows the location of *all* cellphone users across all cellphone networks.

In Section 2, we discuss related work in the area of k-anonymity and location privacy. In Section 3, we present our system and threat model. We introduce our solution and present some remaining challenges in Section 4.

2. RELATED WORK

Gruteser and Grunwald [9] propose both spatial cloaking and temporal cloaking of location information. In the former case, a user's location is cloaked such that there are at least $k-1$ other users in the cloaked area. In the latter case, sending of a query to a location-based service is delayed until at least $k-1$ other users have visited the user's location. In this paper, we concentrate on spatial cloaking, but our approach also applies to temporal cloaking. Gruteser and Grunwald have a trusted "location anonymizer" perform the cloaking based on a quadtree. Upon a query, the location anonymizer subdivides space into quadrants until it finds a quadrant that contains the query issuer and fewer than $k-1$ other users. The parent quadrant becomes the cloaked area. Gedik and Liu [6] let users have personalized values of k, and the cloaked area corresponds to the minimum bounding rectangle of k users. Mokbel et al. [13] observe that this approach can leak information about a user's location (e.g., some users will be on the boundary of the rectangle). They use a balanced quadtree that is traversed bottom-up for better performance until a quadrant with at least k users is found. In our approach, we choose the bottom-up strategy and allow users to personalize k.

Beresford [2] finds that, if a location-based service is familiar with the cloaking algorithm and knows the locations of all users within the cloaked area, the service could infer the identity of the query issuer from the shape of the cloaked area. Namely, this happens when the cloaked area generated for the query issuer is different from the cloaked areas that would have been generated for the other users in the cloaked area. Kalnis et al. [10] and Bettini et al. [3] later re-discover this finding. Kalnis et al. and Mascetti and Bettini [12] present (centralized) cloaking algorithms that are not susceptible to this attack. In our approach, we leave it up to a user to decide what kind of cloaking algorithm to use. She can use either an algorithm similar to Mokbel et al.'s algorithm that does not necessarily guarantee her privacy, but is easy to compute, or an algorithm similar to Mascetti and Bettini's that is robust in terms of privacy, but more expensive.

Chow et al. [4] propose the first distributed approach for location k-anonymity. A user who wants to access a location-based service broadcasts a message with Bluetooth or WiFi. Nearby users respond to this message with their current location. If the number of responses is smaller than $k-1$, the user repeats the process, but has the nearby users forward the message, maybe iteratively. The user then computes her cloaked location and, for increased privacy, asks a nearby user to send her query for the cloaked location to the location-based service. Ghinita et al. [8] show that this approach often fails to achieve location privacy, since the query issuer tends to be in the center of the cloaked area. The same authors [7] later show that their earlier approach can be slow and propose an approach based on a distributed hash table instead. Here, a user knows at least the positions of the two users that immediately follow and precede her in the hash table. Furthermore, for robustness reasons, a user also needs to know the positions of $log_2(n)$ other users, where n is the number of users. In summary, the proposed distributed approaches for location k-anonymity have the drawback that nodes can learn location information about other nodes, so the nodes have to trust each other [7].

3. SYSTEM AND THREAT MODEL

In this section, we present our system and threat model.

3.1 System Model

We present our system model in Figure 1. The figure omits actual location-based services, which a user would access once she learns that at least $k-1$ other users are in her area, likely via a proxy or an anonymous communication network (e.g., Tor [5]) to hide her identity from the service. For scalability reasons, there are multiple *coverage areas*, where a coverage area corresponds to the area covered by a particular instantiation of our system (e.g., a city or a province). A coverage area is divided into a well-defined grid of equally sized, square cells. The width of a cell is chosen such that, for most cells, there is a realistic chance that multiple users can be located in the cell. For example, a cell could have a width of 100 meters. Moreover, there are four kinds of parties: location brokers, users, secure comparison servers, and a directory server.

A location broker keeps track of the current location (i.e., the current cell) of a *subset* of the users in the coverage area. There are multiple location brokers, each keeping track of the location of a *different* subset of users, with the intersection of any two subsets being empty. Each broker is maintained by a different organization. For example, the operator of a cellphone network could maintain a location broker that keeps track of the location of the operator's customers in the coverage area. A location broker does not necessarily provide coverage for all cells in the coverage area. For example, whereas a broker maintained by a cellphone network operator would likely cover most cells, a broker operated by the provider of a WiFi network would provide coverage only for a subset of the cells.

Users carry a mobile device (e.g., a cellphone or a laptop) with them that can locate itself (e.g., using GPS or nearby WiFi base stations). A user registers her current location (i.e., her current cell) with exactly one of the location brokers, where she can choose with which one. Likely, if the provider of the communication service exploited by the user's device runs a location broker, the user will (maybe implicitly) register her location with this broker, since the provider already knows or at least has an estimate of the user's location. We assume that users always register their location with a broker. This assumption is also made in the earlier work. More registrations will lead to smaller cloaked areas, which in turn will increase the quality of service obtained from a location-based service and will give users an incentive to register. If a location broker is run by the provider of the user's communication service, registering her location continuously does not lead to additional loss of privacy for the user, since the provider already has this information.

A secure comparison server interacts with a user to let the user learn whether there are at least k users who have registered the user's current cell as their location across *all*

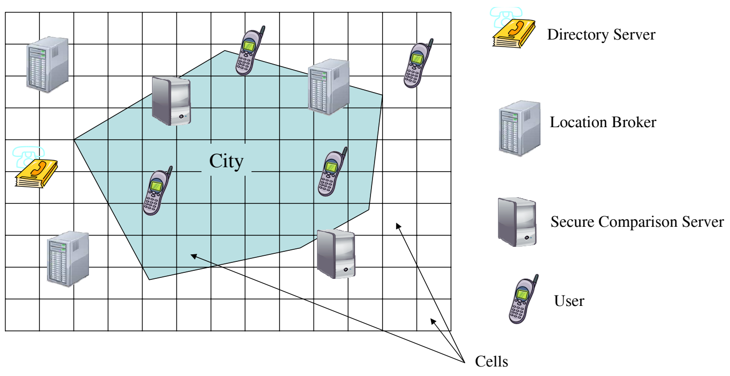

Figure 1: System model. A user register her location with a location broker, whose contact information is provided by the directory server. The user can learn whether there are at least k registered users in her cell by contacting all location brokers and one of the secure comparison servers.

location brokers. (See Section 4 for the detailed protocol.) Each secure comparison server is maintained by a different organization. An organization can maintain both a location broker and a secure comparison server. A secure comparison server provides coverage for the entire coverage area.

The directory server publishes contact information for the location brokers and for the secure comparison servers in the coverage area. Moreover, it publishes coverage information for location brokers, that is, which broker provides coverage for which cells in the coverage area. This way, users can choose a location broker to register with and a secure comparison server to run our protocol with.

3.2 Threat Model

In our threat model, the location brokers and the secure comparison servers are honest-but-curious, that is, they honestly follow our protocol, but are curious about learning location information.

A location broker can learn the location (i.e., cell) of users who register their location with this particular broker. Accordingly, a broker can learn the number of users in a cell that have registered this cell as their location with this particular broker. However, a location broker should not learn the *total* number of users in a cell that have registered this cell as their location across *all* location brokers. Knowing this number makes possible tracking attacks, similar to the one presented in Section 1. Similarly, a location broker should not learn the location of users who register their location with any other location broker. A location broker can learn the identity of a user when she registers with the broker; this assumption is also made in the earlier work, and some brokers already have this information (e.g., an operator of a cellphone network). We assume that the organizations that run the location brokers do not collude with each other. Legal means (e.g., privacy laws or a contract between a user and a location broker) can enforce this assumption. Technical enforcement means make less sense here, since today's cellphone network operators know their customers' location and identity and could potentially share this information with each other. For the same reason, we assume that location brokers do not collude with users.

A secure comparison server should learn neither a user's location nor the total number of registered users in a cell. This implies that the server should not learn the individual numbers for each location broker, either (except using back channels if a secure comparison server is run by the same organization as a location broker). A secure comparison server might collude with other secure comparison servers to learn users' location. Due to the same reason mentioned above, we assume that secure comparison servers do not collude with location brokers (except in the implicit case where a broker and a server are run by the same organization, here it can learn at most the location and number of users registered with this broker).

A user should learn only her own location and whether the number of people in her cell (or superset of cells) is at least k, where k is a value of her choice. A user carries only one mobile device with her, and the device faithfully reports its location to a broker. A user cannot register multiple times with a single broker at the same time, since the broker authenticates the user's device. All these assumptions are also made in the earlier work. A user might still try to register multiple times with *different* brokers at the same time. This could let other users erroneously conclude that k-anonymity holds. Finally, a user might collude with a secure comparison server to learn the number of people in her cell (or superset of cells).

The directory server should not learn any location information about users. The server might misbehave, for example, it might list a location broker multiple times as providing coverage for a single cell, or it might track clients by providing them different information.

4. DISTRIBUTED K-ANONYMITY

In this section, we describe how a user learns whether there are at least k users in her area. We then list several challenges that remain to be addressed.

4.1 Secure Greater Than

The goal of a user is to learn whether there are at least k registered users (including herself) in the user's *query area*, where k is a value chosen by the user and where the query area initially corresponds to the user's current cell. If the user learns that there are fewer than k users in this cell, she can enlarge the query area to a superset of cells that contains the user's current cell and re-execute the protocol for the enlarged area. This process can be repeated multiple times. As mentioned in Section 2, a user can choose between different types of enlargement algorithms to determine the query area, which lets her trade off between privacy and cost. A user registers her current cell as her location with exactly one of the location brokers, but there is no need for the user to register additional cells when enlarging the query area.

Our protocol uses the techniques of public-key cryptography, but we require the cryptosystem used to have a special algebraic property: that it is *additive homomorphic*. An additive homomorphic cryptosystem is one in which, given only $\mathcal{E}(m_1)$ and $\mathcal{E}(m_2)$, where $\mathcal{E}(m)$ is an encryption of message m, one can efficiently compute $\mathcal{E}(m_1 + m_2)$. There are several cryptosystems that fulfill this property, for example, the Paillier cryptosystem [14].

To learn whether there are at least k users in her query area, a user first needs to identify the location brokers that provide coverage for (maybe parts of) the query area. The user must not ask the directory server for a list of brokers that provide this coverage, else the server could learn the user's location. Instead, the user should download the entire directory (or recent changes to it) from the server on a regular basis, such as once a day. The directory is signed, which allows retroactive detection of misbehaviour by the directory server. Another option is to have multiple directory servers, where users accept information from a server only if it is signed by a threshold of the servers, similar to the directory servers in Tor.

The user then interacts with the relevant location brokers and a secure comparison server, as illustrated in Figure 2. The user first asks each broker covering the query area for the number of users who have registered a cell in the query area as their current location with this particular broker. A broker gives this number to the user in a covert way (i.e., the user cannot learn it). Then, the user sums up the received numbers without learning the sum and, with the help of one of the secure comparison servers, determines whether this sum is at least k. Note that a user should not directly connect to a location broker (except to the broker where she is registered) to learn the number of registered users. Information about this connection (e.g., the user's IP address) might allow the broker to re-identify the user and to learn her location from her query. Instead, the user should proxy her communication through the broker that she is registered with or through an anonymous communication network.

Let us now discuss our protocol in detail. In the first stage, the user contacts each relevant location broker. A contacted broker chooses a secure comparison server, l, among the set of secure comparison servers listed by the directory server. All contacted brokers need to choose the same server for a particular user in a particular run of our protocol. If there are v_j users in the query area who have registered with broker j, broker j encrypts v_j with public key C_l, as published by the directory server, and sends $\mathcal{E}_{C_l}(v_j)$ to the user. The user then calculates $\mathcal{E}_{C_l}(r + \sum_i v_i)$ using the additive homomorphic property of the encryption scheme, where r is a random number generated by the user that will keep the total number of users hidden from secure comparison server l.

In the second stage, the user sends $\mathcal{E}_{C_l}(r + \sum_i v_i)$ and $\mathcal{E}_{C_l}(r + k)$ to secure comparison server l, which decrypts and compares the two values and informs the user of the result. Since both the sum and k are obscured with r, the server can learn neither of them.

Our protocol gracefully deals with crashes of a location broker or of a secure comparison server. In the first case, the user contacts the remaining brokers, which might still report a sufficient number of registered users. To work around the second case, we can let the brokers choose a set of candidate secure comparison servers, instead of only a single one. Over time, the directory server will learn of the crash of a location broker or of a secure comparison server and will remove it from the directory.

4.2 Challenges

Our protocol looks simple at first sight. However, there remain several challenges that need to be addressed.

Challenge 1: Sending $\mathcal{E}_{C_l}(r+\sum_i v_i)$ and $\mathcal{E}_{C_l}(r+k)$ to secure comparison server l is problematic, because it might reveal the total number of users to a secure comparison server and a location broker that are run by the same organization. Assume that the location broker is the only broker that covers the query area. Here, based on the knowledge of $\sum_i v_i$ (where the sum covers only one broker), the broker and the server can jointly determine r, which allows them to compute k. In turn, once they know a user's k, the server and the broker can infer the total number of registered people in any query area chosen by the user, as long as the user's choice of k is static and the query area is covered by the broker. The coverage condition guarantees that the broker will be contacted by the user and hence can learn the query area. Otherwise, the server and the broker could learn only the total number of people, but not for which query area. To avoid these information leaks, we need to hide the user's input to the comparison, $r + k$, from the secure comparison server. Moreover, we also need to hide the result of the comparison from the server, else the server could still infer $r + k$ in case it is found to be equal to $r + \sum_i v_i$.

Challenge 2: Using binary search, a user might be able to learn the precise number of users in a cell. Namely, the user could present $\mathcal{E}_{C_l}(r+\sum_i v_i)$ multiple times to the secure comparison server, maybe in re-randomized form or with a different value of r each time. By adjusting the value of k in each run of the protocol, the user can perform a binary search for the actual value of $\sum_i v_i$. Previous research has not considered this attack. In the traditional approach with a central trusted server, this server learns the query area and the identity of a user, which could allow the server to detect the attack. In our scenario, this is more difficult for the secure comparison server since the query area remains hidden from it.

Challenge 3: Consistent with earlier work, our threat model assumes that a location broker can detect attempts by a user to register with the broker multiple times in parallel. Having multiple location brokers, as it is the case in our solution, introduces a new vulnerability. Namely, a user could register multiple times, but each time with a *different* location broker. This way, other users might be wrongly told that their k-anonymity preference is satisfied.

Challenge 4: The location brokers need to pick one of the secure comparison servers. We cannot let a user choose a server because the user might pick a colluding server, which might allow her to learn the total number of registered users in her query area. Instead, we need to choose a secure comparison server such that, over time, the risk of a user colluding with a server is limited by k/n, where n corresponds to the number of secure comparison servers, with k of them colluding with the user. Therefore, we cannot statically as-

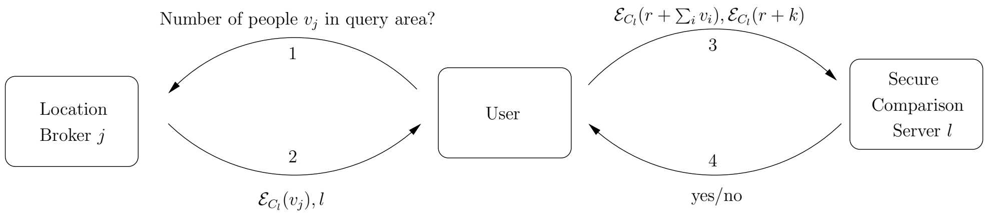
Number of people v_j in query area?
1
Location Broker j
2
$\mathcal{E}_{C_l}(v_j), l$
User
$\mathcal{E}_{C_l}(r + \sum_i v_i), \mathcal{E}_{C_l}(r + k)$
3
Secure Comparison Server l
4
yes/no

Figure 2: Distributed k-anonymity protocol. C_l is the public key of secure comparison server l. $\mathcal{E}_{C_l}(\cdot)$ denotes encryption with public key C_l, where the cryptosystem is additive homomorphic.

sign a secure comparison server to a user, since we might be unlucky and pick a colluding one. Moreover, a user might decide not to trust servers maintained by particular organizations, and she might refrain from using our system if we forced her to use such a server all the time.

5. CONCLUSIONS

We have presented a protocol for location privacy based on k-anonymity that requires neither a single trusted server nor users to trust each other. There remain several challenges that need to be addressed. We present solutions for these challenges in a technical report [16].

Acknowledgements

We thank the anonymous reviewers for their comments. This work is supported by the Natural Sciences and Engineering Research Council of Canada.

6. REFERENCES

[1] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid. In *Proceedings of 17th International World Wide Web Conference (WWW2008)*, pages 237–248, April 2008.

[2] A. R. Beresford. Location privacy in ubiquitous computing. Technical Report 612, Computer Laboratory, University of Cambridge, January 2005.

[3] C. Bettini, S. Mascetti, X. S. Wang, and S. Jajodia. Anonymity in Location-based Services: Towards a General Framework. In *Proceedings of 8th International Conference on Mobile Data Management (MDM 2007)*, pages 67–79, May 2007.

[4] C.-Y. Chow, M. F. Mokbel, and X. Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In *Proceedings of 14th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'06)*, pages 171–178, November 2006.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of 13th USENIX Security Symposium*, pages 303–319, August 2004.

[6] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of 25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 620–629, June 2005.

[7] G. Ghinita, P. Kalnis, and S. Skiadopoulos. MobiHide: A Mobile Peer-to-Peer System for Anonymous Location-Based Queries. In *Proceedings of Proceedings of 10th International Symposium on Spatial and Temporal Databases (SSTD 2007)*, pages 221–238, July 2007.

[8] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVÉ: Anonymous Location-Based Queries in Distributed Mobile Systems. In *Proceedings of 16th International World Wide Web Conference (WWW2007)*, pages 371–380, May 2007.

[9] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, pages 31–42, May 2003.

[10] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preserving Anonymity in Location Based Services. Technical Report TRB6/06, School of Computing, The National University of Singapore, 2006.

[11] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz. AnonySense: Opportunistic and Privacy-Preserving Context Collection. In *Proceedings of 6th International Conference on Pervasive Computing (Pervasive 2008)*, pages 280–297, May 2008.

[12] S. Mascetti and C. Bettini. A Comparison of Spatial Generalization Algorithms for LBS Privacy Preservation. In *Proceedings of International Workshop on Privacy-Aware Location-based Mobile Services (PALMS)*, May 2007.

[13] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proceedings of 32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 763–774, September 2006.

[14] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proceedings of EUROCRYPT '99*, pages 223–238, May 1999.

[15] H. Shin, V. Atluri, and J. Vaidya. A Profile Anonymization Model for Privacy in a Personalized Location Based Service Environment. In *Proceedings of 9th International Conference on Mobile Data Management (MDM'08)*, pages 73–80, April 2008.

[16] G. Zhong and U. Hengartner. A Distributed k-Anonymity Protocol for Location Privacy. Technical Report CACR 2008-17, Centre for Applied Cryptographic Research, University of Waterloo, September 2008.