A Privacy-Preserving in Wireless Sensor Networks using generalization and suppression methods

B.Karthika[#1],J.K.Jeevitha*[2]
Assistant Professor,
Department of Information Technology,
PSNA College of Engineering and Technology
Dindigul,
Tamilnadu,
India.
[1]karthikabm@gmail.com
[2]selvajeeva31@gmail.com

***Abstract*— Monitoring personal locations with a potentially untrusted server poses privacy threats to the monitored individuals. This paper provides a formal presentation of combining generalization and suppression to achieve *k*-anonymity. Generalization involves replacing a value with a less specific but semantically consistent value. Suppression involves not releasing a value at all. The Preferred Minimal Generalization Algorithm (MinGen), which is a theoretical algorithm presented herein, combines these techniques to provide *k*-anonymity protection with minimal distortion.**

Index Terms— **Location privacy, wireless sensor networks, location monitoring system.**

I. INTRODUCTION

Today's globally networked society places great demand on the collection and sharing of person-specific data for many new uses [1]. This happens at a time when more and more historically public information is also electronically available. When these data are linked together, they provide an electronic image of a person that is as identifying and personal as a fingerprint even when the information contains no explicit identifiers, such as name and phone number.
Other distinctive data, such as birth date and postal code, often combine uniquely[2] and can be linked to publicly available information to re-identify individuals. So in today's technically-empowered data rich environment, how does a data holder, such as a medical institution, public health agency, or financial organization, share person-specific records in such a way that the released information remain practically useful but the identity of the individuals who are the subjects of the data cannot be determined? One way to achieve this is to have the released information adhere to *k*-anonymity [3]. A release of data is said to adhere to *k*-anonymity if each released record has at least (*k*-1) other records also visible in the release whose values are indistinct over a special set of fields called the quasi-identifier [4]. The quasi-identifier contains those fields that are likely to appear in other known data sets. Therefore, *k*-anonymity provides privacy protection by guaranteeing that each record relates to at least *k* individuals even if the released records are directly linked (or matched) to external information.

This paper provides a formal presentation of achieving *k*-anonymity using generalization and suppression. G*eneralization* involves replacing (or recoding) a value with a less specific but semantically consistent value. *Suppression* involves not releasing a value at all. While there are numerous techniques available2, combining these two offers several advantages. First, a recipient of the data can be told what was done to the data. This allows results drawn from released data to be properly interpreted. Second, information reported on each person is "truthful" which makes resulting data useful for fraud detection, counter-terrorism surveillance, healthcare outcome assessments and other uses involving traceable person-specific patterns3. Third, these techniques can provide results with guarantees of anonymity that are minimally distorted. Any attempt to provide anonymity protection, no matter how minor, involves modifying the data and thereby distorting its contents, so the goal is to distort minimally. Fourth, these techniques can be used with preferences a recipient of the released data may have, thereby providing the most useful data possible. In this way, algorithmic decisions about how to distort the data can have minimal impact on the data's fitness for a particular task. Finally, the real-world systems Datafly [5] and m-Argus [6], which are discussed in subsequent sections, use these techniques to achieve *k*-anonymity. Therefore, this work provides a formal basis for comparing them.

II SYSTEM MODEL

Figure 1depicts the architecture of our system, where there are three major entities, *sensor nodes*, *server*, and *system users*. We will define the problem addressed by our system, and then describe the detail of each entity and the privacy model of our system

Problem definition: Given a set of sensor nodes s1; s2; : : : ; sn with sensing areas a1; a2; : : : ; an, respectively,a set of moving objects o1; o2; : : : ; om, and a required anonymity level k, (1) we find an aggregate location for each sensor node si in a form of Ri = (Areai ;Ni), where Areai is a rectangular area containing the sensing area of a set of sensor nodes Si and Ni is the number of objects residing in the sensing areas of the sensor nodes in Si, (2) we build a spatial histogram to answer an aggregate query Q that asks about the number of objects in a certain area Q:Area based on the aggregate locations reported from the sensor nodes.

Sensor node: Each sensor node is responsible for determining the number of objects in its sensing area, blurring its sensing area into a cloaked area A, which includes at least k objects, and reporting A with the number of objects located in A as aggregate location information to the server. We do not have any assumption about the network topology, as our system only requires a communication path from each sensor node to the server through a distributed tree [8]. Each sensor node is also aware of its location and sensing area.

Server: The server is responsible for collecting the aggregate locations reported from the sensor nodes, using a spatial histogram to estimate the distribution of the monitored objects, and answering range queries based on the estimated object distribution. Furthermore, the administrator can change the anonymized level k of the system at anytime by disseminating a message with a new value of k to all the sensor nodes.

System users: Authenticated administrators and users can issue range queries to our system through either the server or the sensor nodes, as depicted in Figure 2. The server uses the spatial histogram to answer their queries.

Privacy model: In our system, the sensor nodes constitute a trusted zone, where they behave as defined in our algorithm and communicate with each other through a secure network channel to avoid internal network attacks, for example, eavesdropping, traffic analysis, and malicious nodes [7], [9]. Since establishing such a secure network channel has been studied in the literature [7],[9], the discussion of how to get this network channel is beyond the scope of this paper. However, the solutions that have been used in previous works can be applied to our system. Our system also provides anonymous communication between the sensor nodes and the server by employing existing anonymous communication techniques [10], [11]. Thus given an aggregate location R, the server only knows that the sender of R is one of the sensor nodes within R. Furthermore, only authenticated administrators can change the k-anonymity level and the spatial histogram size. In emergency cases, the administrators can set the k-anonymity level to a small value to get more accurate aggregate locations from the sensor nodes, or even set it to zero to disable our algorithm to get the original readings from the sensor nodes, in order to get the best services from the system. Since the server and the system user are outside the trusted zone, they are untrusted.

Since our system only allows each sensor node to report a k-anonymous aggregate location to the server, the adversary cannot infer an object's exact location with any delity. The larger the anonymity level, k, the more difficult for the adversary to infer the object's exact location. With the k-anonymized aggregate locations reported from the sensor nodes.This is a nice privacy-preserving feature, because the object count of a small area is more likely to reveal personal location information. The dentition of a small area is relative to the required anonymity level, because our system provides lower quality services for the same area if the anonymized level gets stricter.

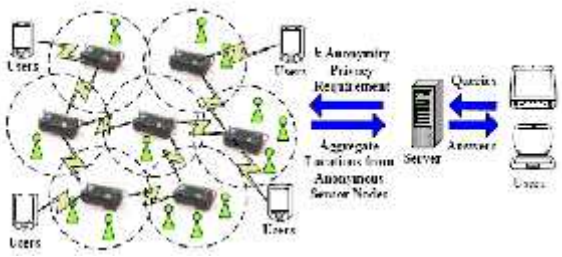

Fig.1:System Architecture.

III ANONYMIZATION METHODS

3.1. Generalization including suppression

The idea of generalizing an attribute is a simple concept. A value is replaced by a less specific, more general value that is faithful to the original. In Figure 2 the original ZIP codes {*02138, 02139*} can be generalized to *0213**, thereby stripping the rightmost digit and semantically indicating a larger geographical area.

In a classical relational database system, domains are used to describe the set of values that attributes assume. For example, there might be a ZIP domain, a

number domain and a *string* domain. We extend this notion of a domain to make it easier to describe how to generalize the values of an attribute. In the original database, where every value is as specific as possible, every attribute is considered to be in a ground domain. For example, 02139 is in the ground ZIP domain, Z0.In order to achieve *k*-anonymity we can make ZIP codes less informative. we do this by saying that there is a more general, less specific domain that can be used to describe ZIPs, say Z1, in which the last digit has been replaced by 0 (or removed altogether). There is also a mapping from Z0 to Z1, such as 02139 $\rightarrow$0213*.

Given an attribute *A*, we say a *generalization for an attribute* is a function on *A*.

That is, each f: $A \rightarrow B$ is a generalization.

$$A_c \xrightarrow{f_0} A_1 \xrightarrow{f_1} \ldots \xrightarrow{f_{n-1}} A_n$$

is a generalization sequence or a functional generalization sequence. Given an attribute *A* of a private table PT, I define a **domain generalization hierarchy** DGHA for *A* as a set of functions fh : h=0,…,n-1 such that:

$$A_o \xrightarrow{f_0} A_1 \xrightarrow{f_1} \ldots \xrightarrow{f_{n-1}} A_n$$

A=$A0$ and $|An|$ = 1. DGHA is over:

$$\bigcup_{h=0}^{n} A_h$$

Clearly, the fh's impose a linear ordering on the Ah's where the minimal element is the ground domain $A0$ and the maximal element is An. The singleton requirement on An ensures that all values associated with an attribute can eventually be generalized to a single value. In this presentation we assume Ah, h=0,…,n, are disjoint; if an implementation is to the contrary and there are elements in common, then DGHA is over the disjoint sum of Ah's and definitions change accordingly. Given a domain generalization hierarchy DGHA for an attribute *A*, if $vi \varepsilon Ai$ and $vj \varepsilon Aj$ then I say $vi \leq vj$ if and only if $i \leq j$ and:

$$f_{j-1}(\ldots f_i(v_i)\ldots) = v_j$$

This defines a *partial ordering* $\leq$ on:

$$\bigcup_{h=0}^{n} A_h$$

Such a relationship implies the existence of a **value generalization hierarchy** VGHA for attribute *A*. Here expand the representation of generalization to include suppression by imposing on each value generalization hierarchy a new maximal element, atop the old maximal element. The new maximal element is the attribute's suppressed value. The height of each value generalization hierarchy is thereby incremented by one. No other changes are necessary to incorporate suppression. Figure 2 and Figure 3 provides examples of domain and value generalization hierarchies expanded to include the suppressed maximal element (*****). In this example, domain Z0 represents ZIP codes for Cambridge, MA, and E0 represents race. From now on, all references to generalization include the new maximal element; and, hierarchy refers to domain generalization hierarchies unless otherwise noted.

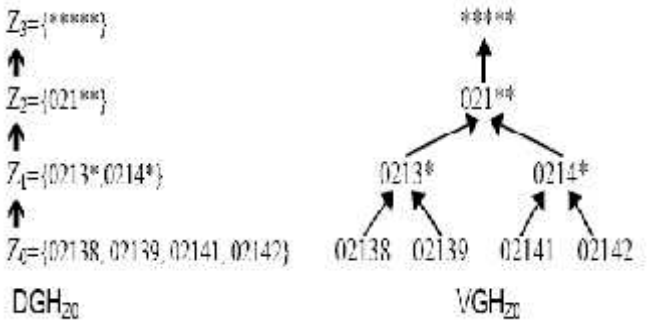

Figure 2 ZIP domain and value generalization hierarchies including suppression

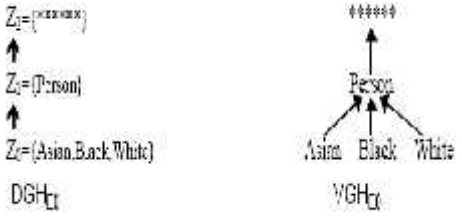

Figure 3 Race domain and value generalization hierarchies including suppression

Given table PT, generalization can be effective in producing a table RT based on PT that adheres to *k*-anonymity because values in RT are substituted with their generalized replacements. The number of distinct values associated with each attribute is non-increasing, so the substitution tends to map values to the same result, thereby possibly decreasing

the number of distinct tuples in RT. Figure 4.a and 4.b are example for generalization table.

A generalization function on tuple *t* with respect to *A1*,…, *An* is a function *ft* on *A1*×…×*An* such that:

$f_t(A_1,\ldots\ldots,A_n)=(f_{t1}(A_1),\ldots\ldots,f_m(A_n))$

where for each *i*: 1,…,*n*, *fti* is a generalization of the value *t*[*Ai*]. The function *ft* is a set function. I say *ft* is generated by the *fti*'s. Given *f*, *A1*,…,*An*, a table *T*(*A1*,…,*An*) and a tuple *tεT*, i.e., *t*(*a1*,…,*an*) g(T)={k.f(t):tεT and|f^{-1}(f(t)) |=k}

The function *g* is a multi-set function and *f*-1 is the inverse function of *f*. We say that *g* is the multi-set function generated by *f* and by the *fi*'s. Further, We say that *g(*T*) is a generalization of table* T. This does not mean, however, that the generalization respects the value generalization hierarchy for each attribute in T. To determine whether one table is a generalization with respect to the value generalization hierarchy of each attribute requires analyzing the values themselves.

Let DGHi be the domain generalization hierarchies for attributes *Ali* where
i=1,…,*An* Let Tl[*Al1*,…,*AlAn*] and Tm[*Am1*,…,*AmAn*] be two tables such that for each
i:1,..,*n*, *Ali*,*Amiε*DGHi. Then, We say table Tm is **a generalization of table** Tl,
written Tl ≤ Tm, if and only if there exists a generalization function *g* such that *g*[Tl] = Tm and is generated by *fi*'s where: *tl*Tl, *ali* ≤ *fi*(*ali*) = *ami and fi* : *Ali*→ *Ami* and each *fi* is in the DGHi of attribute *Ali*. From this point forward, I will use the term *generalization* (as a noun) to denote a generalization of a table.

Race E_0	ZIP Z_0
Black	02138
Black	02139
Black	02141
Black	02142
White	02138
White	02139
White	02141
White	02142

PT

Race E_1	ZIP Z_0
Person	02138
Person	02139
Person	02141
Person	02142
Person	02138
Person	02139
Person	02141
Person	02142

$GT_{[1,0]}$

Race E_1	ZIP Z_1
Person	0213*
Person	0213*
Person	0214*
Person	0214*
Person	0213*
Person	0213*
Person	0214*
Person	0214*

$GT_{[1,1]}$

Figure 4.a Examples of generalized tables for PT

3.2. Algorithm for finding a minimal generalization with minimal distortion

The algorithm presented in this section combines these formal definitions into a theoretical model against which real-world systems will be compared.

Figure 5 presents an algorithm, called MinGen, which, given a table
PT(*Ax*,…,*Ay*), a quasi-identifier QI={*A1*,…,*An*}, where {*A1*,…,*An*} Í {*Ax*,…,*Ay*}, a *k*-anonymity constraint, and domain generalization hierarchies DGHAi, produces a table **MGT** which is a *k*-minimal distortion of **PT**[*QI*]. It assumes that *k* < |**PT**|, which is a necessary condition for the existence of a *k*-minimal generalization

Race F_0	ZIP Z_2
Black	021**
Black	021**
Black	021**
Black	021**
White	021**
White	021**
White	021**
White	021**

$GT_{[0,2]}$

Race F_0	ZIP Z_1
Black	0213*
Black	0213*
Black	0214*
Black	0214*
White	0213*
White	0213*
White	0214*
White	0214*

$GT_{[0,1]}$

Figure 4.b Examples of generalized tables for PT

The steps of the MinGen algorithm are straightforward. [step 1] Determine if the original table, PT, itself satisfies the *k*-anonymity requirement; and if so, it is the *k*-minimal distortion. In all other cases execute step 2. [step 2.1] Store the set of all possible generalizations of PT over QI into *allgens*. [step 2.2] Store those generalizations from *allgens* that satisfy the *k*-anonymity requirement into *protected*. [step 2.3] Store the *k*-minimal distortions (based on *Prec*) from *protected* into *MGT*. It is guaranteed that |*MGT*| ³ 1. [step 2.4] Finally, the function *preferred()* returns a single *k*-minimal distortion from *MGT* based on user-define specifications.

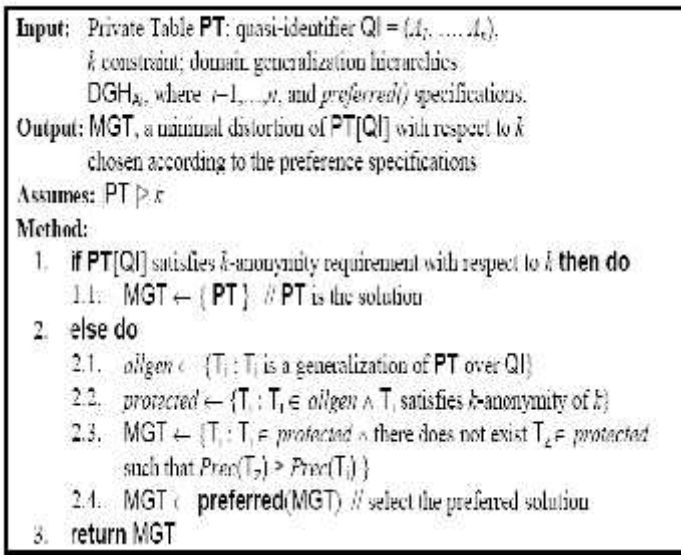

Figure5 Examples of generalized tables for PT

IV CONCLUSION

In this paper, we propose a privacy-preserving location monitoring system for wireless sensor networks. We design two in-network location anonymization methods generalization and suppression methods.

REFERENCES

1 L. Sweeney, Information Explosion. *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, L. Zayatz, P. Doyle, J. Theeuwes and J. Lane (eds), Urban Institute, Washington, DC, 2001.

2 L. Sweeney, *Uniqueness of Simple Demographics in the U.S. Population*, LIDAPWP4. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA: 2000. Forthcoming book entitled, *The Identifiability of Data*.

3 L. Sweeney. *k*-Anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (7), 2002.

4 T. Dalenius. Finding a needle in a haystack – or identifying anonymous census record. *Journal of Official Statistics*, 2(3):329-336, 1986.

5 L. Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly system. *Proceedings, Journal of the American Medical Informatics Association*. Washington, DC: Hanley & Belfus, Inc., 1997.

6 A. Hundepool and L. Willenborg. m- and t-argus: software for statistical disclosure control. *Third International Seminar on Statistical Confidentiality*. Bled: 1996.

7. M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, .Privacy-aware location sensor networks,. in *Proc. of HotOS*, 2003.

8. D. Culler and M. S. Deborah Estrin, .Overview of sensor networks, . *IEEE Computer*, vol. 37, no. 8, pp. 41.49, 2004.

9 A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, .SPINS: Security protocols for sensor netowrks,. in *Proc. of MobiCom*, 2001.

10 J. Kong and X. Hong, .ANODR: Anonymous on demand routing with untraceable routes for mobile adhoc networks,. in *Proc. Of MobiHoc*, 2003.

11 P. Kamat, Y. Zhang,W. Trappe, and C. Ozturk, .Enhancing source location privacy in sensor network routing,. in *Proc. of ICDCS*,