Turk J Elec Eng & Comp Sci, Vol.18, No.2, 2010, © TÜBİTAK
doi:10.3906/elk-0907-120

k-anonymity based framework for privacy preserving data collection in wireless sensor networks

Hayretdin BAHŞİ, Albert LEVİ
National Research Institute of Electronics and Cryptology,
Gebze, İzmit-TURKEY
e-mail: bahsi@uekae.tubitak.gov.tr
Faculty of Engineering and Natural Sciences, Sabancı University,
İstanbul-TURKEY
e-mail: levi@sabanciuniv.edu

Abstract

In this paper, k-*anonymity notion is adopted to be used in wireless sensor networks (WSN) as a security framework with two levels of privacy. A base level of privacy is provided for the data shared with semi-trusted sink and a deeper level of privacy is provided against eavesdroppers. In the proposed method, some portions of data are encrypted and the rest is generalized. Generalization shortens the size of the data transmitted in the network causing energy saving at the cost of information loss. On the other hand, encryption provides anonymization with no information loss and without saving energy. Thus, there is a tradeoff between information loss and energy saving. In our system, this tradeoff is intelligently managed by a system parameter, which adjusts the amount of data portions to be encrypted. We use a method based on bottom up clustering that chooses the data portions to be encrypted among the ones that cause maximum information loss when generalized. In this way, a high degree of energy saving is realized within the given limits of information loss. Our analysis shows that the proposed method achieves the desired privacy levels with low information loss and with considerable energy saving.*

Key Words: *Anonymity; Wireless Sensor Networks; Data Privacy.*

1. Introduction

Anonymity is defined as a subject being not identifiable within a set of subjects [1]. For many years, the anonymity problem referred to hiding sender and/or receiver identities of messages in data and communication networks. DC-Net and Mix-net solutions were proposed for this aim [2, 3]. Many other practical anonymity solutions were presented for ISDN networks [4], web and e-mail applications in the Internet [5, 6], for anonymous routing [7], and for ad hoc networks [8]. On the other hand, there are few studies about anonymity problem in WSNs, mostly based on trying to hide the location or time information of events. The current literature on anonymity in WSNs is given in Section 6 of this paper.

In the context where the entities have several attributes, such as in databases and data mining, the problem of privacy poses several challenges that cannot be easily solved by stripping off identity information, due to the fact that other data fields and sources may be jointly used to deduce private information. Suppose several organizations need to share their electronic information, such as public health or demographic data. However, they want to protect the privacy of their consumers or personnel during this information-sharing operation. Simply stripping off the name or social security number from the data set does not solve the privacy problem. It is still possible to identify the owner of a record by using this reduced data in conjunction with other data from various public information sources. This attack technique is called a 're-identification attack' [9] or "record linkage attack" [10]. In order to prevent these type of attacks, Samarati and Sweeney proposed k-anonymity [9]. Basically, k-anonymity brings a specific restriction to the anonymity problem such that one subject is specifically targeted to be hidden among k-1 other subjects. In other words, the attributes that may help to identify a subject are modified, via k-anonymization, in such a way that each subject has an anonymity set having a size of at least k-1. Generally, in privacy problems, the owner of the record or individual having the attributes in the record is assumed to be the subject of anonymity. k-anonymization of data is performed by suppression or generalization of some parts. These generalization and suppression operations, however, cause information loss. Thus, it is important to minimize the amount of loss by minimizing the number of suppression and generalization operations, while keeping the data k-anonymous. It is shown that achieving optimal k-anonymization by minimum number of suppressions is NP-hard even when the alphabet size of attributes equal to three [11].

k-anonymity tries to solve the privacy preserving data sharing problem based on a trusted data collection model [10]. In this model, data is collected from data owners by a data publisher. The data publisher releases the data to a data recipient, who analyzes it to complete a required task. A generic example is that of hospitals sharing their medical records with medical research institutions. In this example, the data owners are patients, the data publishers are hospitals, and the data recipients are medical research institutions. Data publishers collect all the details of patient records, but they are required to share them by obeying some privacy criterion. Therefore, patients have to fully trust hospitals in completing privacy preserving operations, but they do not trust the data recipient. Data publishers generally do not know the details of the analysis that is performed by the data recipient, so they share information as much as possible. In the literature, this problem is called "privacy preserving data publishing" [10].

As a result of advances in sensor and wireless technology, wireless sensor networks (WSNs) have emerged as an important information gathering system over wide areas. WSNs have been used to perform critical missions in uncontrolled environments. In some sensor network applications, sensor nodes may deployed over territory that is not physically protected, and may be required to exchange data without a fixed routing structure, gather various types of information from uncontrolled areas, and transfer them to the sinks in controlled or semi-controlled areas. The collected information is not always an aggregated value like average temperature or humidity; it may also be information about specific individuals or events, so that the privacy of each event information becomes important. Moreover, the data gathered in these and other sensor network applications may contain several attributes for an entity. This could enable the launching of re-identification attacks, even in the case where the identities are withheld. Therefore, in such a situation, a k-anonymity based solution to provide privacy in WSNs would be more effective.

One of the major threats in wireless mediums is the eavesdropper threat. During the transmission of gathered data to the sink or central server, an adversary having eavesdropping capabilities can sniff the network and obtain event information. Since the sensed area is uncontrolled, the adversary can use his own systems to collect extra information from the area, combine his knowledge with the sniffed event information and thereby determine attributes of specific events, such as location and time. In WSN applications of enemy tracing, habitat monitoring, traffic monitoring and human tracking [12], eavesdropping threats have to be dealt with so that the privacy of data during the transmission is ensured.

The privacy problem is not limited to the threat of eavesdroppers. In some situations, the data shared with the sink or central server has to fulfill some privacy requirements. There may be a threat of sink capture in some WSN environments, where physical security is not guaranteed. Physical capture of the sink affects the whole system, since it stores all event information.

In other applications, the sink itself may not be considered by WNS users to be a fully trustable entity. For example, consider a wireless body area network system where the patient's health status are centrally tracked by a central server (i.e. the sink) in the hospital. Users may not want the central server to know their exact spatio-temporal information unless an emergency occurs. Therefore, privacy of personal information shared with the sink is also needed. As another example, people using WSN applications, such systems caring for the elderly or smart home monitoring systems, may need to protect their privacy from the parties with whom they share personal information.

The trusted data collection model used in privacy preserved data publishing methods [10] does not fit directly into WSN environment since the data shared entity, or sink, may not be fully trusted and there may be other un-trusted parties like eavesdroppers. Anonymization must be conducted across multiple trusted entries so that the compromising a trusted entity does not lead to the loss of all the data of the system. Thus, a new data collection model for WSN environments must be be adopted.

One of the main design criteria in WSNs is energy consumption, requiring designers of privacy preserved data collection systems for WSNs to concentrate on reducing energy costs. Studies show that [13], the energy consumption is heavily dependent on transmission/reception of data packets. Therefore, reducing the size of event information plays a crucial role in energy conservation.

1.1. Our contribution

In this study, we propose a framework for the problem of privacy preserved data collection in WSN applications. Based on the threat model, trusted and un-trusted parties are appropriately chosen among the WSN components and k-anonymity is adapted to this environment.

As discussed in section 2, our threat model is based on the threat due to untrusted eavesdroppers and semi trusted sinks. Therefore, the model has two privacy criteria: $k1$-anonymity for the data received by the semi trusted sink and $k2$-anonymity for the data transmitted in the network that can be captured by the untrusted eavesdropper, where $k2 \geq k1$.

In this paper, the k-Anonymization Clustering Method (k-ACM) is proposed to provide two-layer privacy in WSNs. Our method minimizes the data loss, while keeping the energy consumption within reasonable limits and satisfying the privacy requirements of the threat model. k-ACM is based on the unweighted pair group method with arithmetic mean (UPGMA) [14], which is a type of bottom-up clustering method. All anonymization stages are formalized in the bottom-up clustering schema. We perform an information loss calculation, anonymity level measurement and vector distance calculation using the notion of entropy in information theory.

Our method first $k1$-anonymizes the data to a base level for semi trusted sinks via generalization. Next, the data is further anonymized against eavesdroppers until the data becomes $k2$-anonymous with the encryption and generalization operations. A trivial solution seems to be totally encrypting the $k1$-anonymized data. If $k2$-anonymity is sufficient as a privacy requirement for the eavesdroppers and some amount of data loss can be tolerated by the sink, instead of fully encrypting the $k1$-anonymous data, this data is $k2$-anonymized via encryption and generalization operations. In this study, it is shown that this partial encryption method considerably decreases the energy consumption by shortening the lengths of messages.

Another contribution of this paper is the presentation of a dynamic taxonomy tree in generalization operations so that smaller amounts of information loss.

The rest of the paper is organized as follows. Section 2 presents the details of the network and the threat model of the proposed framework. In Section 3, some basic definitions and background information about UPGMA method and generalization operations with static taxonomy trees are provided. In Section 4, generalization with dynamic taxonomy trees, the information loss metric and the proposed k-anonymization technique k-ACM are presented. In this section we also analyze the effects of k-ACM on the size of data. In Section 5, experimental results that show the effectiveness of the proposed method are given and the results are discussed. Section 6 describes related work in the literature. Section 7 concludes the paper.

2. Network and threat model

Wireless sensor networks are generally deployed in open areas. Third parties can determine some attributes of detected events using their own sensors or by directly observing events. Record linkage attacks may also be conducted to identify the event owner. Our threat model is based on providing privacy by preventing "record linkage attacks" during data collection.

Privacy threats are both due to eavesdroppers and the Sink, thus, our threat model addresses the privacy requirements of both of these two threat types. In our model, the privacy requirement levels of the system against the sink and against the eavesdroppers are not the same. We employ a privacy mechanism in which there are two privacy levels associated with the eavesdroppers and sink: *untrusted* eavesdroppers and *semi-trusted* sink.

Untrusted eavesdroppers are assumed to be capable of capturing data, but the system should ensure that the privacy of the intercepted data is be protected to some extent. In this way, the eavesdroppers can only learn limited information from the intercepted data.

On the other hand, while the *semi-trusted* sink is allowed to legally obtain data, this data should also have a specific privacy property. However, the privacy protection level of the data that the sink obtains is lower than that of the eavesdroppers. In this way, the sink can learn more detailed information in comparison to eavesdroppers, but the detail of the information obtained is still limited.

In our network model, there is one sink and a number of sensor nodes. Some sensor nodes serve as *aggregation nodes*, where all anonymization operations on the data takes place. Therefore, our model protects the event data between aggregation nodes and the sink.

To prevent "record linkage attack", k-anonymity can be provided by fusing different events. Therefore, trusted entities at which anonymizations can occur must be determined. Due to threats of WSN environments, these fuse points must be distributed so that the compromising of one point does not lead to the compromising

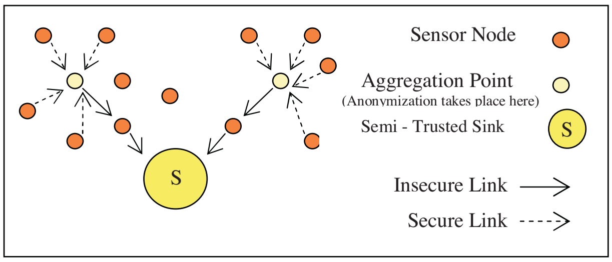


Figure 1. Visualization of network and threat models.

of the entire network data. Also, it is convenient to choose points as close as possible to the sensors. Therefore, aggregation nodes act as locally trusted parties for the corresponding local sensors. Our anonymization framework solves the privacy problem for the data travelling from aggregation nodes to sink. In Figure 1, the basics of the network and threat models are shown. The links between sensor nodes and aggregation nodes are assumed to be secure. Here, to ensure confidentiality of the traffic between sensor nodes and aggregation nodes, an appropriate key management mechanism [15, 16] should be employed. On the other hand, the links between aggregation nodes and semi-trusted sink are assumed to be insecure, an issue that is addressed in this paper.

We assume that our WSN uses widely accepted data-centric routing Protocols, such as SPIN [17] or directed diffusion [18], for finding appropriate routes from sensors to sink.

3. Basic definitions and background

In this section, we first explain some basic definitions that are used in the paper. Some background information about static taxonomy trees and the UPGMA algorithm are also given.

3.1. Basic definitions

Quasi-identifier Attribute: An attribute that is not able to identify a subject by itself, but may help to identify subject with a combination of similar attributes. The set of all quasi-identifier attributes of table T is Q.

***k*-anonymity:** Suppose that $T(Q)$ refers to the new table produced by keeping the quasi-identifier attributes and removing the others from table T. T has the k-anonymity property if and only if each record is indistinguishable from other k-1 records in $T(Q)$. Generalization and suppression are more common techniques for making the data k-anonymous.

Anonymity Set: If a subject cannot be discriminated from a set of other subjects, the set is called an anonymity set of that subject. In k-anonymity, each subject has an anonymity set having at least k-1 elements.

Generalization: The generalization operation replaces the value of a quasi-identifier attribute with a more general value. For example, a birth date such as '04.05.1977' can be replaced by '1977' in a generalization operation. Numerical attribute values may also be generalized to numeric intervals.

Suppression: Deletion of a quasi-identifier attribute of a record or removing the whole record entirely.

For example, assume that a WSN is constructed for traffic monitoring. This application collects information about the vehicles passing through various locations of the city. Attributes of the sample data are given

Table 1. Attributes of sample data.

Attribute Name	Attribute Type	Values of Attribute
Vehicle Type	categorical	train, truck, bus, pickup, vans, car
Time	numerical	Values between 00:00 and 24:00
Location	categorical	Serin Street, Buket Street, Selvi Street, Mimoza Street, Durmaz Street

Table 2. Sample data for a traffic monitoring application.

Vehicle Type	Time	Location
car	12:05	Buket Street
train	13:00	Selvi Street
bus	12:50	Serin Street
pickup	11:30	Serin Street
bus	12:30	Durmaz Street
truck	12:20	Selvi Street

in Table 1. A sample set of data is given in Table 2. Suppose that k is chosen as two and the data in Table 2 is made 2-anonymous. A sample 2-anonymized version of the data using only generalization operations is shown in Table 3. In this example, the privacy problem is assumed to be the prevention of a record linkage attack and all attributes are considered to be quasi-identifiers. Taxonomy trees for vehicle type and location information are given in Figure 2 and Figure 3, respectively.

3.2. Taxonomy trees

A taxonomy tree is a tree structure that is created for each categorical quasi-identifier attribute to replace an existing attribute value with more general one in the k-anonymization process [19, 20, 21]. This replacement is a generalization operation. Leaves of the tree contain distinct values of attributes and nodes in the higher levels of the tree contain more general attribute values. During the anonymization, replacement is done with the values in the higher levels of the tree. There is a root of the taxonomy tree. If the attribute value is generalized up to this point, then the attribute value has no information. Suppression is considered to be another operation for anonymization in the literature [20], but in fact it is a generalization operation where the attribute value is generalized to the root of attribute's taxonomy tree. Consider a sensor network that collects location information, such as address information. A possible taxonomy tree for this location attribute can be constructed as in Figure 3. Suppose that k is chosen as 2 and the location attribute is the quasi-

Table 3. Anonymized version of the sample data.

Vehicle Type	Time	Location
normal sized vehicle	11:30-12:05	Istasyon Avenue
high sized vehicle	12:20-13:00	Selvi Street
bus	12:30-12:50	Tuzla
normal sized vehicle	11:30-12:05	Istasyon Avenue
bus	12:30-12:50	Tuzla
high sized vehicle	12:20-13:00	Selvi Street

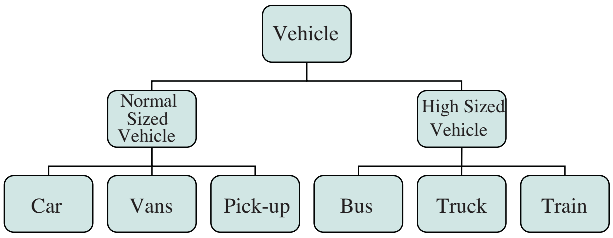


Figure 2. Static taxonomy tree for vehicle type.

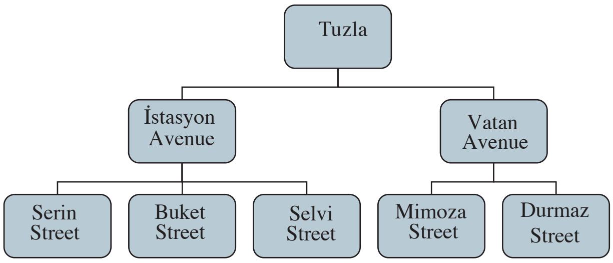


Figure 3. Static taxonomy tree for location information.

identifier. If there are two records having the location information 'Buket Street' and 'Selvi Street', and if they are anonymized to a common value, the value of the location attribute is generalized to common ancestor in the tree, which in this case is actually 'Istasyon Avenue'. The values of the location attribute for two records are replaced with this more general attribute value. Thus, no one can discriminate these two records from each other using location information.

3.3. UPGMA method

UPGMA [14] The unweighted pair group method with arithmetic mean (UPGMA) is based on the idea of iteratively joining the two closest clusters until one cluster is left. A suitable distance definition is required to measure the distance between any two clusters. All distances between each pair of clusters, which are computed according to the distance definition, are stored in a distance matrix at each iteration. At the beginning, each input vector is considered to be an individual cluster. The closest two clusters are found and combined into one common cluster. The distance of the newly formed cluster to the other clusters are recalculated and the distance matrix is updated. This procedure is repeated until one cluster is formed.

4. Proposed k-anonymization method

In this section, our work in k-ACM is explained. The first subsection details the ideas behind the proposed dynamic taxonomy tree for generalization methods. The second subsection introduces the information loss metric that is used in selecting the suitable portions for encryption operations and evaluating k-ACM results. The third subsection presents the proposed algorithm k-ACM. The fourth subsection explains the information theoretic method used to measure the k-anonymity level of data. The fifth subsection analyzes the effects of k-ACM on the size of data and gives the formulation for saving energy with k-anonymization.

Table 4. Anonymized version of the sample data.

Vehicle Type	Time	Location
car-pickup	11:30-12:05	Buket Street - Serin Street
train-truck	12:20-13:00	Selvi Street
bus	12:30-12:50	Serin Street - Durmaz Street
car-pickup	11:30-12:05	Buket Street - Serin Street
bus	12:30-12:50	Serin Street - Durmaz Street
train-truck	12:20-13:00	Selvi Street

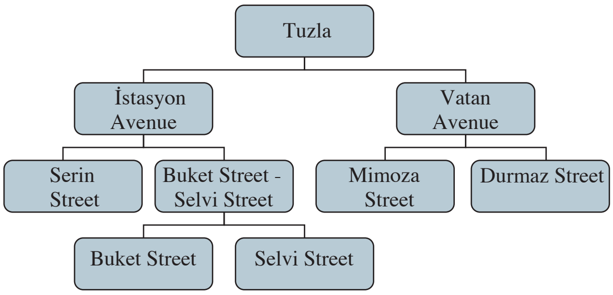


Figure 4. A sample node addition in dynamic taxonomy tree.

4.1. Generalization method with dynamic taxonomy tree

In privacy preserved data collection, the main aim is to share as much as possible with the related parties under the required privacy criterion. Data collection methods generally use static taxonomy trees. Over-generalization is a potential problem of using static taxonomy trees in the generalization of categorical attributes. For example, in the static taxonomy tree given in Figure 3, the generalization of the values 'Buket Street' and 'Selvi street' yields the value 'Istasyon Avenue'. However, this causes information loss, since records containing Istasyon Avenue may also, for example, include 'Serin Street'. To solve this over-generalization problem as much as possible, we propose using a *dynamic taxonomy tree* instead of a static one. In the proposed dynamic taxonomy tree model, the tree is dynamically updated by creating new internal nodes (i.e. attribute values) during the generalization and in an on-demand manner, depending on the nature of data and the required generalization. In this method, a new node is generated when the existing parent node has child(ren) other than the generalized nodes. The newly generated node covers the attribute values of the generalized ones only. In this way, generalization is performed with minimum information loss. Let us continue with the previous example. As shown in Figure 4, in our dynamic generalization approach, 'Buket Street' and 'Selvi Street' cause a new categorical value with name 'Buket Street - Selvi Street' to be generated instead of generalizing to existing value, 'Istasyon Avenue'. This new value ensures that the attribute is either 'Buket Street' or 'Selvi Street', but not 'Serin Street'.

If we apply the dynamic taxonomy tree method to the sample data given in Table 2, the 2-anonymized output in Table 4 is obtained. From this anonymized data, the number of vehicles in each street and the total number of vehicles in each type can be calculated accurately. These calculations cannot be accurately done from the anonymized data in Table 3.

To perform generalization among any of the attribute values using the proposed dynamic taxonomy tree concept, a flexible data structure should be employed to represent the attribute values. In our method, a bit string is employed as this data structure. If an attribute is categorical, the size of the bit string is equal to the total number of elements in the set of attribute values. In this structure, each bit corresponds to a distinct attribute value. To specify which value that attribute has, the corresponding bit of the attribute value is set to one, while the other bits are zero. Bit strings of original data (i.e. data before generalization) have a single '1' bit.

In this data structure, generalizations are implemented by setting the corresponding bits of the attribute values that will be generalized to '1'. Therefore, the total number of bits having the value '1' increases as generalizations occur. A bit string with many bits of value '1' actually represents an internal node.

For a numerical attribute, the range of the attribute can be divided into intervals in which each interval has an equal range suitable size. The size of a numerical attribute's bit string is set to the number of intervals. Each interval corresponds to a distinct bit and if an attribute belongs to an interval, the corresponding bit of the interval is set to one in the bit string. The number of intervals can be determined according to the accuracy requirement for that attribute. A higher accuracy requirement implies a larger number of intervals. Increasing the number of intervals enlarges the size of the messages, and thus the required transmission energy. Therefore, a balance between energy and accuracy must be constructed in choosing the number of intervals.

Suppose the input data is a table T having m attributes and n records. T_{ij} represents the j'th attribute of the i'th record where $\{i : 1 \leq i \leq n\}$ and $\{j : 1 \leq j \leq m\}$. Table T is represented by a set of bit strings B, where B_{ij} is the bit string representation of the j'th attribute of the i'th record. The k'th bit of B_{ij} is written as $B_{ij}(k)$. Suppose that j'th attribute of the table is categorical and there are d_j distinct values. These values are indexed by k and written as $V_j(k)$, where $\{k : 1 \leq k \leq d_j\}$. A bit string of this categorical attribute has a size of d_j and is formed as follows:

$If\ T_{ij} = V_j(k)\ then\ B_{ij}(k) = 1\ else\ B_{ij}(k) = 0\ as\ \forall k :\ 0 \leq k \leq d_j,$

If an attribute is numerical, the range of the attribute is divided into equal-sized intervals. Assume that the j'th attribute is numeric and the attribute range is divided into an e_j number of intervals. Each interval is indexed by k. The bit string representation of this numeric attribute has a size of e_j and is formed as follows:

$If\ T_{ij}\ intersects\ with\ k'th\ interval,\ then\ B_{ij}(k) = 1\ else\ B_{ij}(k) = 0\ as\ \forall k :\ 0 \leq k \leq e_j$

In our proposed model, sensor nodes send their findings directly to aggregation nodes. Aggregation nodes convert quasi-identifier attributes of data to bit strings and k-ACM makes them k-anonymous. Through the k-anonymization process of an attribute, k-ACM uses the notion of dynamic taxonomy trees. During the formation of a dynamic taxonomy tree, the bit string of the newly created internal node of a dynamic taxonomy tree is found by the logical OR operation of bit strings of all child nodes.

4.2. Information loss metric

Calculating the data loss of k-anonymous data is needed to predict the performance of our proposed method under different k-anonymity parameters. In our study, we use the entropy concept of information theory to measure the information loss [22]. The entropy difference between the k-anonymous data and the original data constitutes the information loss. Suppose that T is the input data set having n records and m attributes, B is the bit string representation of this data set, as discussed in Section 4.1, and C is a random variable representing the probability of an attribute value in a k-anonymous data entry being the actual attribute value in the original data. Assume that all the entries of B are normalized according to the number of bits having a value of '1' in that entry (i.e., a 'true bit) and that the normalized version forms the data set $\overline{B}$. A sample

Table 5. A sample bit string representation set.

Records	B_{i1}	B_{i2}	B_{i3}
T_1	00010	01000	10000
T_2	01100	11100	01111

Table 6. A sample normalized version of the bit string representation set.

Records	$\overline{B_{i1}}$	$\overline{B_{i2}}$	$\overline{B_{i3}}$
T_1	00010	01000	10000
T_2	$0\frac{1}{2}\frac{1}{2}00$	$\frac{1}{3}\frac{1}{3}\frac{1}{3}00$	$0\frac{1}{4}\frac{1}{4}\frac{1}{4}\frac{1}{4}$

data set is shown in Table 5. Here, there are two records, and each record has three attributes. Each attribute is categorical and has five distinct attribute values. Table 6 shows the normalized version of the data. During normalization, each entry is divided by the number of true bits in the corresponding bit string entry.

Information loss of a data table T, $IL(T)$, is equal to the conditional entropy, $H(C \mid B)$. Here, the conditional entropy indicates the uncertainty of the prediction for the original attribute values of a record when we have knowledge of the corresponding k-anonymous bit strings of that record. The original data has only one true bit in each bit string, because each original data entry corresponds to one attribute value. However, in k-anonymous data, each entry may have more than one attribute value and thus each attribute value is represented by an additional bit. Therefore, if an entry has only one true bit, then it does not have information loss. In this situation, we can be certain that this true bit is the true bit that comes from the original data. As the number of true bits increases, the disorder of the data increases, because it becomes harder to predict which one is the original true bit. Prediction gets harder because information is lost as a result of the increase in the number of true bits. Conditional entropy, which is used to calculate the disorder of the data, is a good measurement tool for information loss. Conditional entropy $H(C \mid B)$, which is equal to the information loss of table T, $IL(T)$, can be found as follows:

$$IL(T) = H(C \mid B) = \sum_{B_{ij} \in B} p(B_{ij}) H(C \mid B = B_{ij}) = -\sum_{B_{ij} \in B} p(B_{ij}) \sum_{k \in \{1..z\}} p(C = k \mid B_{ij}) \log p(C = k \mid B_{ij}) \tag{1}$$

In Eq. (1), it is assumed that each attribute is converted to bit strings having size z. This implies that all categorical attributes have z distinct attribute values and all numerical attributes have z number of interval ranges. Also, it is assumed that all k's, where the equalities of $p(C = k \mid B_{ij}) = 0$ are true, are excluded from the summation. The random variable C can take on values from the set $\{1..z\}$. In fact, $\overline{B}$ is calculated for finding the value of this random variable.

$$p(C = k \mid B = B_{ij}) = \overline{B}_{ij}(k) \; for \; each \; k : \; 1 \leq k \leq z \tag{2}$$

In Eq. (1), it is assumed that each record has an equal probability of being chosen and that each attribute of the record has the same probability.Therefore, the probability mass function of the j'th attribute of the i'th record, $p(B_{ij})$, is calculated as $p(B_{ij}) = \frac{1}{m.n}$. Eq. (1) can be rewritten as follows:

$$IL(T) = H(C \mid B) = -\sum_{B_{ij} \in B} \frac{1}{m.n} \sum_{k \in 1..z} \overline{B}_{ij}(k). \log \overline{B}_{ij}(k) \tag{3}$$

Suppose that F is the array that contains the number of true bits of the bit string array B. The total number of true bits in B_{ij} is F_{ij}. The total number of elements in $\overline{B}_{ij}(k)$ that has the value $\frac{1}{F_{ij}}$ is equal to F_{ij}, and the rest is zero. Therefore, the second summation operation of Eq. (3) yields the value $\log \frac{1}{F_{ij}}$. The simplest equation for the information loss of the data table T, $IL(T)$, can be calculated as follows:

$$IL(T) = H(C \mid B) = -\sum_{F_{ij} \in F} \frac{1}{m.n} \log \frac{1}{F_{ij}} = \frac{1}{m.n} \sum_{F_{ij} \in F} \log F_{ij} \tag{4}$$

4.3. Clustering algorithm for k-anonymization

k-anonymization produces an output such that the number of any quasi-attribute value set is not less than k. An optimal method must find groups of records possessing this property with minimum data loss. Since this problem has been shown to be an NP-Hard problem [11, 23], various heuristic methods have been developed to prevent data loss as much as possible [9, 21].

In this paper, we adapt a bottom-up hierarchical clustering technique to the k-anonymization problem. The proposed method is called the k-anonym Clustering Method (k-ACM). The clustering algorithm is applied to the portion of the data containing only quasi-identifier attributes. The basic idea is to partition the data vectors into clusters, where each cluster has at least k vectors. After clustering, the vectors in one cluster are anonymized to a common vector. Each cluster forms a representative vector that is actually the k-anonymization output of all vectors in that cluster. All quasi-identifier attributes of the input data are replaced with the corresponding attributes of the representative vector. This clustering process ensures that similar vectors are grouped in clusters so that their anonymization does not cause significant data loss.

The run time of k-ACM is found to be $O(n^2 logn)$, where n is the number of event records. The details of the run time derivation is given in Section 7.Section 4.3.1 describes the method of distance calculation and Section 4.3.2 explains the proposed method. The notation used in these sections is given in Table 7.

4.3.1. Method of distance calculation

The aim of our method is to minimize information loss, while providing the required level of k-anonymity. At each iteration of k-ACM, two clusters are combined. Each cluster combination leads to some generalization operations and therefore to information loss. k-ACM should choose the most suitable cluster pair that creates the least information loss when combined. To do so, a suitable distance calculation method is required . In Section 4.2, the notion of conditional entropy is used to calculate the overall information loss of k-anonymized data. This notion is adapted to calculate the distance between any two clusters as the entropy loss caused by merging them.

At the h^{th} iteration, the distance between the s^{th} and t^th clusters is defined as $D^h[s][t]$. Suppose that the cluster resulting from merging the clusters s and t is represented as cluster u in iteration $h+1$. Cluster

Table 7. Notation table for section 4.3.

Notation Explanation	Notation
i^{th} Aggregation node	A_i
Event information collected in the i^{th} aggregation node	E_i
Iteration Number	h
Input data	T
i^{th} record of input data, T	T_i
Array of clusters at the h^{th} iteration	L^h
s^{th} cluster in L^h where $\{s : 0 < s < L^h\}$	L^h_s
Total number of clusters at the h^{th} iteration	$\|L^h\|$
The array of input vectors belonging to cluster L^h_s	V^h_s
k^{th} bit string of j^{th} input vector of array V^h_s	$V^h_s[j][k]$
Number of input vectors of cluster (size of cluster),L^h_s	$\|V^h_s\|$
Representative vector of cluster L^h_s in bit string	R^h_s
i^{th} bit string of representative vector, R^h_s	$R^h_s[i]$
Number of true bits of bit string, x	$F(x)$
Bit string generation function, which gets two bit strings, x and y and produces the generalization of these strings	$G(x, y)$
Distance matrix at the h^{th} iteration	D^h
Distance value between s^{th} and t^{th} cluster at the h^{th} iteration	$D^h[s][t]$
Information loss occurred during the formation of cluster, L^h_u (Suppose that s^{th} and t^{th} clusters are combined, form the cluster u)	$I^h_u(I^h_u = D^h[s][t])$

u has $|V^h_s + V^h_t|$ number of elements. The merging operation means that $|V^h_s|$ number of input vectors having a value R^h_s and $|V^h_t|$ number of input vectors having a value R^h_t is converted to $|V^h_s + V^h_t|$ number of vectors having a value R^{h+1}_u. The value of the conditional entropy before merging is represented as E^h_{st} and is computed using Eq. (4) as follows:

$$E^h_{st} = \frac{1}{m.(|V^h_s| + |V^h_t|)}|V^h_s| \sum_{i\in\{1..m\}} \log(F(R^h_s[i])) + |V^h_t| \sum_{i\in\{1..m\}} \log(F(R^h_t[i])) \tag{5}$$

E^{h+1}_u is the value of the conditional entropy after merging and is calculated as follows:

$$E^{h+1}_u = \frac{1}{m.(|V^h_s| + |V^h_t|)}(|V^h_s| + |V^h_t|) \sum_{i\in\{1..m\}} \log(F(R^{h+1}_u[i])) = \frac{1}{m} \sum_{i\in\{1..m\}} \log(F(R^{h+1}_u[i])) \tag{6}$$

The distance between cluster s and t at iteration h, $D^h[s][t]$, is:

$$D^h[s][t] = E^{h+1}_u - E^h_{st} \tag{7}$$

4.3.2. k-ACM method

Our proposed method, k-ACM, starts with an initialization phase in which a new cluster is created for each input vector. After initialization, the clustering operation begins and is performed in two distinct stages, the

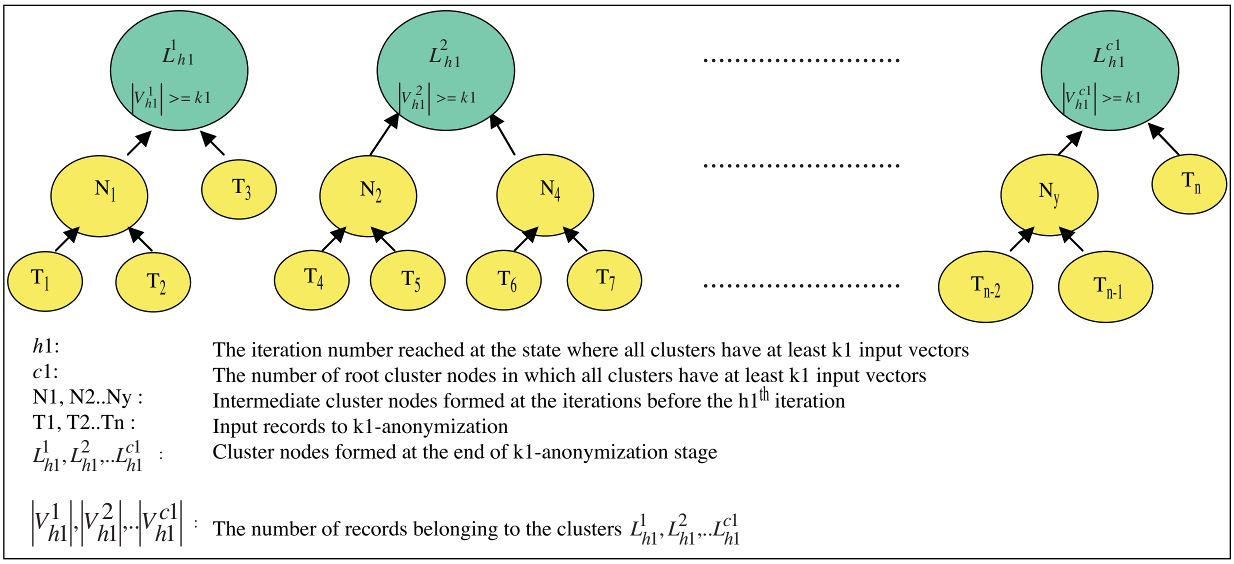
```
Function ClusterCombination:
Input: parameter $k$ for anonymization Output:One combined cluster
  Find the minimum distance from the distance matrix $D^h$, which is $D^h[s][t]$
  Combine the $s^{th}$ and $t^{th}$ clusters and create a new cluster $u$
  Add the elements of $s^{th}$ and $t^{th}$ clusters to the elements of cluster $u$, size of $u$ is $|V_u^{h+1}| = |V_s^h| + |V_t^h|$
  Find the representative vector of the new cluster, $R_u^{h+1}$ by ORing $R_s^h$ and $R_t^h$
  Remove the $s^{th}$ and $t^{th}$ clusters from array of clusters
  Calculate the distance of cluster $u$ to the other clusters, update The distance matrix $D^h$ by removing the distance information
  of $s^{th}$, $t^{th}$ clusters and adding the $u^{th}$ cluster
  $D^{h+1} = D^h$ and $h = h + 1$
   $h1 = h$ and $c1 = |L^h|$ ($h1$ is the iteration number reached at the end of the $k1$-anonymization stage and $c1$ is the number of
  clusters in the array of clusters)
k-ACM (Main Method):
Input: Table $T$ (it is assumed that data has only quasi-identifier attributes), number of records $n$,
number of attributes $m$, anonymization parameter $k1$, anonymization parameter $k2$ ($k2 \geq k1$), output enlargement factor, $M$
Output:$k$-anonymized table $k$-ACM($T$)
1. Initialization
   $h = 1$($h$ is iteration variable) and create array of clusters, $L^1$, with $n$ initial clusters $L^1 = \{L_1^1, L_2^1, ...L_n^1\}$
   Add record, $T_i$ to the input vector array of cluster, $V_i^1$, and $|V_i^1|$, for all $i$ where $\{i : 0 < i < n\}$
   $|L^1| = n$ and initialize the representative vectors to input vectors, $R_i^1 = T_i$
   Initialize the distance matrix $D^1$ by computing all distances according to Eq. (7)
2. $k1-$anonymization
   while not for each cluster $|V_i^h| \geq k1$
         Function ClusterCombination ($k1$)
3. $k2-$anonymization
   while not for each cluster $|V_i^h| \geq k2$
         Function ClusterCombination ($k2$)
4. Output enlargement for partial encryption
   Initialize the set of vectors to be sent to sink, $S$, by the array of clusters,$L_1^{h2}..L_{c2}^{h2}$($|S| = c2$)
   Calculate the maximum number of representative vectors, $\varphi$ that will be sent in the anonymous output: $\varphi = c2 + M.(c1 - c2)$
   while not $|S| = \varphi$
         Select the node, $f$, with the maximum information loss in $S$
         Find the child nodes, $g$ and $h$, of node $f$ from intermediate cluster array, modify $S$ by replacement of node $f$ with
            nodes $g$ and $h$
5. Form the output of $k$-ACM
   For each node in $S$
         If cluster represented by the node has size lower than k2, encrypt the representative vector of that cluster and append it
            to the output with the cluster size value
         Otherwise, append the representative vector of cluster in clear text to the output with the cluster size value
```

Figure 5. k-ACM Algorithm.

$k1$-anonymization and $k2$-anonymization stages. The algorithm for k-ACM is given in Figure 5. The notation used in the algorithm is given in Table 7.

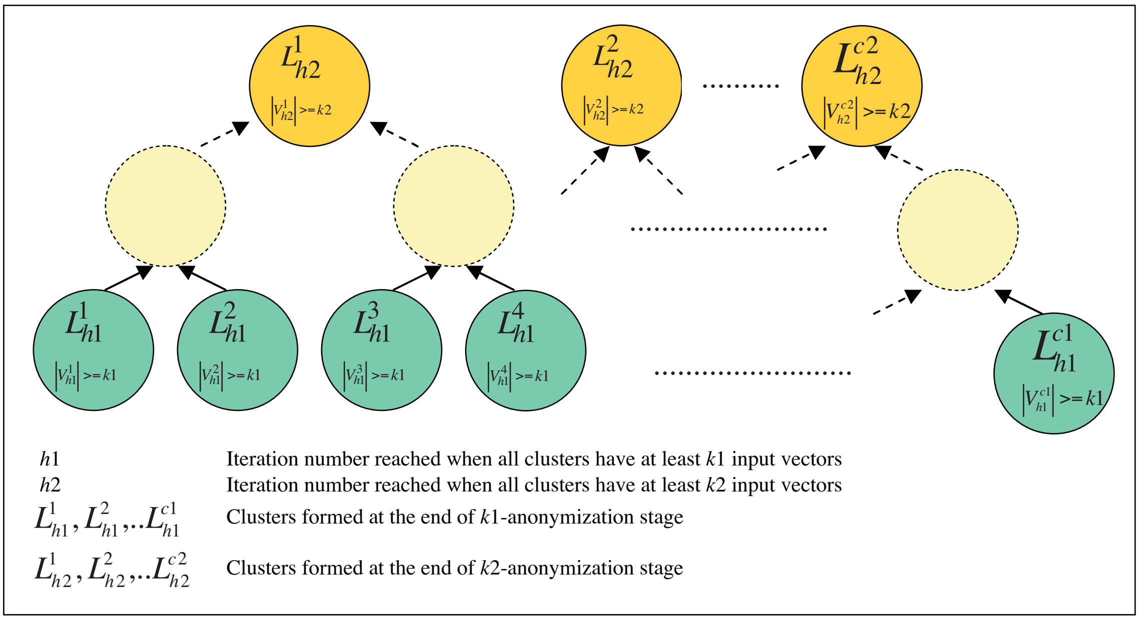
In the $k1$-anonymization stage (2nd item in Figure 5), k-ACM forms the clusters in a bottom-up fashion like UPGMA until each cluster represented has at least $k1$ records. A sample tree structure of the clusters obtained at the end of the $k1$-anonymization stage is shown in Figure 6. Here, $h1$ is defined as the number of iterations needed to complete the $k1$-anonymization stage. In this tree structure, each tree node represents a cluster and the cluster size is the number of records that belong to the corresponding cluster. This tree has $c1$ root nodes, identified as $L_1^{h1}..L_{c1}^{h1}$, and their sizes are at least $k1$.

In the $k2$-anonymization stage (3rd item in Figure 5), bottom-up clustering starts with the clusters represented by the root nodes of the $k1$-anonymization tree and continues until all the root nodes of the tree have sizes of at least $k2$, where $k2 \geq k1$. A tree structure obtained at the end of the $k2$-anonymization stage is shown in Figure 7. Here, $h2$ is defined as the number of iterations after which all the root nodes have at least $k2$ items and bottom-up clustering is completed. This tree structure has $c1$ leaf nodes, $L_1^{h1}..L_{c1}^{h1}$ and $c2$



L^1_{h1} $|V^1_{h1}| >= k1$
L^2_{h1} $|V^2_{h1}| >= k1$
L^{c1}_{h1} $|V^{c1}_{h1}| >= k1$
N1
N2
N4
Ny
T1
T2
T3
T4
T5
T6
T7
Tn
Tn-2
Tn-1
$h1$: The iteration number reached at the state where all clusters have at least k1 input vectors
$c1$: The number of root cluster nodes in which all clusters have at least k1 input vectors
N1, N2..Ny : Intermediate cluster nodes formed at the iterations before the h1th iteration
T1, T2..Tn : Input records to k1-anonymization
$L^1_{h1}, L^2_{h1}, ..L^{c1}_{h1}$: Cluster nodes formed at the end of k1-anonymization stage
$|V^1_{h1}|, |V^2_{h1}|..|V^{c1}_{h1}|$: The number of records belonging to the clusters $L^1_{h1}, L^2_{h1}..L^{c1}_{h1}$

Figure 6. A sample tree structure of clusters obtained at the end of the $k1$-anonymization stage.



L^1_{h2} $|V^1_{h2}| >= k2$
L^2_{h2} $|V^2_{h2}| >= k2$
L^{c2}_{h2} $|V^{c2}_{h2}| >= k2$
L^1_{h1} $|V^1_{h1}| >= k1$
L^2_{h1} $|V^2_{h1}| >= k1$
L^3_{h1} $|V^3_{h1}| >= k1$
L^4_{h1} $|V^4_{h1}| >= k1$
L^{c1}_{h1} $|V^{c1}_{h1}| >= k1$
$h1$ Iteration number reached when all clusters have at least $k1$ input vectors
$h2$ Iteration number reached when all clusters have at least $k2$ input vectors
$L^1_{h1}, L^2_{h1}, ..L^{c1}_{h1}$ Clusters formed at the end of $k1$-anonymization stage
$L^1_{h2}, L^2_{h2}, ..L^{c2}_{h2}$ Clusters formed at the end of $k2$-anonymization stage

Figure 7. A sample tree structure of clusters obtained at the end of the $k2$-anonymization stage.

root nodes, $L^{h2}_1..L^{h2}_{c2}$. In each cluster combination operation, the closest clusters are found and a new cluster, which contains all the vectors belonging to the closest clusters found, is formed. Distance calculations are done according to Eq. (7). The vector representative of the new cluster is a bitwise OR of the representative vectors of child nodes. Here, the OR operation acts as a generalization operation and the clusters in higher tree levels have more generalized representative vectors. A sample case for the cluster combination operation is shown in Figure 8. Suppose that the closest clusters in the hx iteration are L^{hx}_1, L^{hx}_2 and their representative vectors

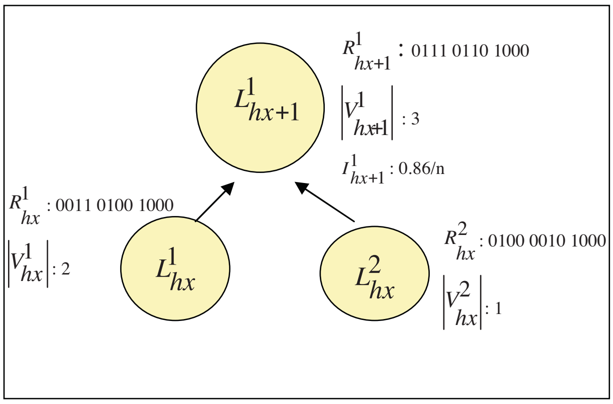


Figure 8. A ample case for forming a new cluster by combining the two closest clusters.

are R_1^{hx}, R_2^{hx}. Assume that representative vectors have three bit strings, each having a length of four bits. This means there are three quasi-identifiers in the data set, of which all are categorical and may have four distinct values. The new cluster is labelled by L_1^{hx+1}. Its representative vector, R_1^{hx+1}, is obtained by ORing R_1^{hx} and R_2^{hx}. Each cluster combination results in information loss due to the increase in the number of true bits. The cluster sizes, L_1^{hx} and L_2^{hx} are represented as $|V_1^{hx}|$ and $|V_2^{hx}|$, respectively. The size of the new cluster, $|V_1^{hx+1}|$, is the summation of the sizes of the child nodes, $|V_1^{hx}|$ and $|V_2^{hx}|$. The distance between L_1^{hx} and L_2^{hx} is stored in the variable, I_1^{hx+1}. Before explaining stage 4 of the k-ACM method in 5, the motivation behind output enlargement and partial encryption is given below. As discussed before, information loss is an important design criterion in k-ACM. Another important criterion is energy savings. Sensor nodes are usually scattered over an area in which there are no power supplies other than a simple battery. Therefore, increasing the battery lifetime is desired in almost all WSN applications. A sensor node consumes energy for different processes like event sensing, CPU processing, or transmitting/receiving data packets. Although the encryption process requires a considerable amount of CPU processing, recent studies [13] show that energy consumption rates for transmission/reception is over three orders of magnitude greater than the energy consumption rates for encryption. Since each sensor node acts as a router for the messages of other nodes and one message travels over many hops in the network, saving energy in transmission/reception operations becomes a crucial design criterion. These facts motivate WSN designers to shorten the length of the packets.

In fact, there is a tradeoff between information loss and energy consumption. We analyze this tradeoff for two extreme cases of the k-ACM method.

1. Make the data $k2$-anonymous via generalization operations at aggregation points and send it to the semi-trusted sink. This case corresponds to the case where the data computed at the end of stage 3 of Figure 5 are sent to the sink.
2. Make the data $k1$-anonymous with generalization operations, encrypt all this anonymous data by a shared key with the semi-trusted sink and send it to the sink. This case corresponds to the case where the data obtained at the end of stage 2 are entirely encrypted.

Both cases fulfill the requirements of our threat model. In the first case, in which only generalization is performed, the length of the data is minimized and the number of encryptions is zeroed. In this way, energy consumption is also minimized. However, due to the generalization operations, information loss is maximized in this case. In the second extreme case, since only encryption is performed, the length of the data transmitted is maximized, yielding maximum energy consumption. However, since the encrypted data will be decrypted at sink, there is no extra information loss in this case.

To cope with the trade-off between energy consumption and information loss more efficiently, we introduce an intelligent partial encryption alternative in k-ACM. In k-ACM, there is an allowance for encryption operations in the $k2$-anonymization stage. Basically, k-ACM uses this allowance for the portions of data that have a high potential for information loss when generalization is applied.

k-ACM can effectively find the appropriate data entries for encryption operations (Stage 4 of the k-ACM method in Figure 5) as described below. It first $k2$-anonymizes the data (Stage 3 of the k-ACM algorithm) as depicted in the tree structure of Figure 7. Let us define the set S as the set of all representative vectors to be sent to the sink. Initially the set S contains $c2$ nodes, which are the root nodes of the tree, $L_1^{h2}..L_{c2}^{h2}$. All of these nodes are $k2$-anonymized. If no encryptions are allowed, the initial content of S is sent to the sink that causes maximum information loss, as discussed above. To reduce information loss, the encrypted versions of the representative vectors with k-anonymization levels less than $k2$ can be sent to sink. This process requires moving down the tree in Figure 7. In other words, some nodes in S are replaced by their children. This is done in an iterative manner until a certain limit is reached. At each iteration, k-ACM chooses the element of S with the highest information loss and replaces it with its child nodes. This replacement increases the size of data sent to sink by one vector, but the quality of data is also increased, since we have now discarded some generalization operations by moving down the tree.

An example is depicted in Figure 9. Assume that the node, L_a^{h2}, has the maximum information loss and the nodes, L_{cx}^{h2-1}, L_{cy}^{h2-1} are its child nodes. L_a^{h2} is replaced by its child nodes and the set S is now composed of the nodes, $L_1^{h2}..L_{cx}^{h2-1}, L_{cy}^{h2-1}..L_{c2}^{h2}$. With this replacement, we can eliminate the information loss that has occurred after merging the nodes $L_{cx}^{h2-1} and L_{cy}^{h2-1}$. Therefore, the quality of the data represented by the new set S is greater than before. However, the length of S is increased by one, and L_{cx}^{h2-1} and L_{cy}^{h2-1} must be encrypted since they are not $k2$-anonymized.

In the 4th stage of the k-ACM algorithm given in Figure 5, the number of replacements is adjusted by a predetermined threshold value, an output enlargement factor, M, where $0 \leq M \leq 1$. This value determines the maximum size of S, which is $c2 + M(c1 - c2)$ vectors. The replacement continues until the size of S reaches this value.

At the end of the replacement process, the vectors of the nodes in S are transmitted to the sink along with the node sizes (Stage 5 of k-ACM). Before transmission, representative vectors of nodes having anonymity levels less than $k2$ are encrypted to make the data $k2$-anonymous for eavesdroppers. The other vectors, which are actually $k2$-anonymous, are sent in cleartext.

In k-ACM, M determines the tradeoff between information loss and energy savings. If M is zero, then $c2$ $k2$-anonymized data entries are sent to the sink and none of them are encrypted. This corresponds to the first extreme case explained above. However, if M is one, then the output includes $c1$ vectors and all of them are encrypted. This corresponds to the second extreme case discussed above. In general, large M values mean greater output size and higher data quality at the cost of greater energy consumption for both communication and encryption. Small M values mean smaller output size and less quality data with the benefit of less energy consumption. The analysis of this tradeoff is given in Section 5.

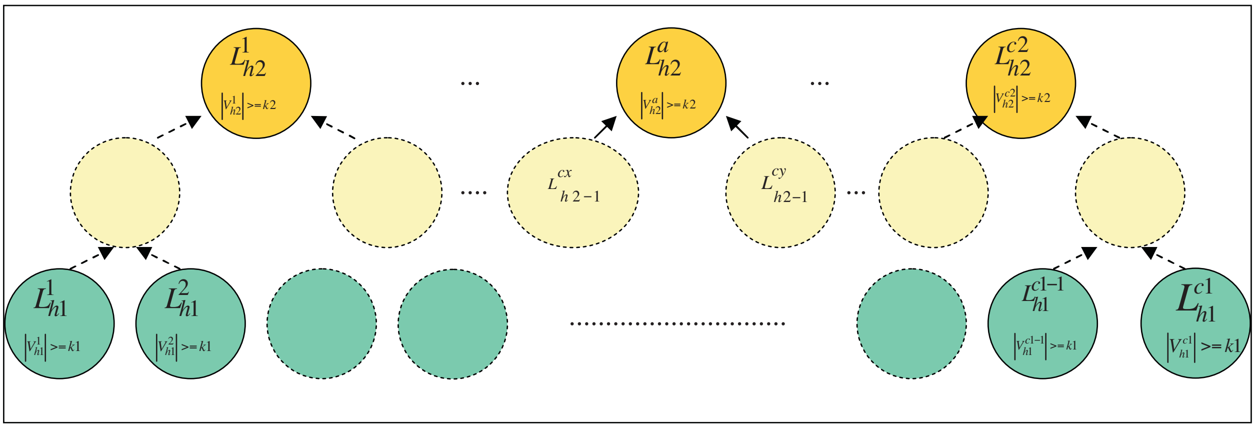


Figure 9. Selection of data entries for encryption.

4.3.3. Network-wide operation of k-ACM

In our network model, there are many aggregation nodes that obtain local event data from regular sensor nodes and forward them to the sink after applying k-ACM to the data. Therefore, the k-ACM algorithm runs at each aggregation node *in parallel*.

Pseudo-code that shows the network-wide operation of k-ACM in a parallel manner is shown in Figure 10.

```
For each aggregation node in parallel
    while time period is not exceeded or buffer of node is not full
        Collect and accumulate local event information from sensors
    Run $k$-ACM
    Send the output of $k$-ACM to the sink
```

Figure 10. Network-wide operation of the k-ACM algorithm in a parallel manner.

4.3.4. Anonymity measurement

k-anonymity guarantees a certain level of anonymity because it ensures that each subject cannot be differentiated from among the other $k-1$ subjects. In this section, we quantify the amount of anonymity provided by k-ACM.

Suppose that A and B are sets of records from the original data and k-anonymous data, respectively. Conditional entropy, $H(A \mid B)$, is used as an anonymity measurement method. $H(A \mid B)$ gives the uncertainty level of prediction for the record in A when the corresponding anonymous version of the record in B is known. Here, more uncertainty means a higher anonymity level. The amount of anonymity, Q, is calculated as follows:

$$Q = H(A \mid B) = \sum_{b \in B} p(b).H(A \mid B = b) = -\sum_{b \in B} p(b). \sum_{a \in A} p(a \mid b). \log p(a \mid b) \qquad (8)$$

where $p(b)$ is the probability mass function of B, and $p(a \mid b)$ is the conditional probability of a value of A, a, given a value of B, b. The lower bound for Q corresponds to the case where the data is exactly k-Anonymous.

In other words, suppose that the anonymous data has n records and that each record has exactly the same set of quasi-identifier records as $k-1$ other subjects. In this situation, for each record b, $p(b)$ is $1/n$. $p(a \mid b)$ is $1/k$ for k records and 0 for the other $n-k$ records. By evaluating Eq. (8) with these values, we can calculate the minimum anonymity level of k-anonymous data, which is denoted as Q_{min}, as follows:

$$Q_{min} = logk \tag{9}$$

However, the purpose of k-ACM is not to form clusters having exactly k elements, but to increase the quality of data as much as possible under the criterion that each cluster must have at least k elements. Therefore, the number of clusters produced by k-ACM is generally less than n/k and the number of elements in each cluster is greater than or equal to k. Suppose that C_F is the final set of clusters, C_F^i is the i^{th} cluster and $s(C_F^i)$ represents the number of elements in cluster C_F^i. Q is computed as follows:

$$Q = -\sum_{C_F^i \in C_F} \frac{1}{s(C_F^i)}.\log\frac{1}{s(C_F^i)} \tag{10}$$

The uncertainty takes on the lowest value when each cluster has exactly k elements. However, the uncertainty increases when clusters have different number of elements and number of clusters decreases. Therefore, the inequality $Q \geq Q_{min}$ holds for every possible output of k-ACM.

4.3.5. *k*-anonymization output size and energy saving

In our study, k-anonymization is considered as a privacy mechanism; however, k-anonymization shortens the length of the event messages as well. In this way, energy consumption is reduced. In other words, k-anonymization makes the quasi-identifier fields of k or more records identical. It is not needed to resend these identical parts repeatedly. These parts must be sent to the sink just once along with the number of occurrences. With this reduction technique, the data is shortened while information about the individual events is conserved.

In section 4.3, it is shown that the k-ACM output is composed of representative vectors of clusters and their sizes. The number of representative vectors is determined by the value of $c2 + M(c1-c2)$, where M is the predetermined threshold value, i.e. the output enlargement factor. This variable can take values from zero to one. As a result, the range for the number of vectors is $[c2, c1]$. Assume that in the data there are n records, m categorical attributes, and each attribute has p distinct attribute values. $k1$, $k2$ are the parameters of the k-anonymization. The size of the clusters cannot exceed $2k_2$, therefore cluster size can be represented in the output with at most $log(2k_2)$bits. The total length of all representative vectors of clusters and their cluster sizes can be at most $(c_2+M(c_1-c_2)).(mp+log(2k_2))$. Originally, the size of the data is $mn\log p$. The decrease ratio, D, is described as the ratio of difference of input and output size to the input size in a k-anonymization operation. D is computed for the overall k-ACM output as follows:

$$D = \frac{mn\log(p) - (c2 + M(c1-c2)).(mp + log(2k_2))}{mn\log(p)} \tag{11}$$

The decrease ratio directly affects the energy consumption by saving certain amount of energy due to the reduced length of data transferred to the sink. The energy saving metric, C_G, is defined as a ratio of the amount of energy saved due to k-anonymization. Using the decrease ratio, D, the energy savings, C_G, is calculated as follows:

$$C_G = 1 - \frac{2.5h_{sregion} + 2.5(1-D)h_{region} + 8.58\ 10^{-4}\beta}{2.5(h_{sregion} + h_{region})} \tag{12}$$

where $h_{sregion}$ is the expected number of hops an event message travels from a sensor node to the aggregation node; h_{region} is the expected number of hops from aggregation points to the sink; D is the decrease ratio; and β is the number of encrypted entries $M(c1-c2)$. Derivations for $h_{sregion}$, h_{region} and C_G are given in detail in Section 7 of the Appendix.

To calculate the energy savings after the $k1$-anonymization stage of the k-ACM algorithm, the decrease ratio at this stage, $D_{k1-anonymization}$, must be calculated. For this calculation Eq. (11) needs to be revised. The number of representative vectors at this stage is $c1$ and the cluster size occupies a bit length of $log(2k_1)$. $D_{k1-anonymization}$ is found as follows:

$$D_{k1-anonymization} = \frac{mn\log(p) - c1.(mp + \log(2k_1))}{mn\log(p)} \tag{13}$$

If the value of D in Eq. (12) is replaced with $D_{k1-anonymization}$, then the energy savings that is guaranteed at the end of the $k1$-anonymization stage can be calculated. The decrease ratio in the $k2$-anonymization stage is calculated as follows:

$$D_{k2-anonymization} = \frac{c1.(mp + \log(2k_1)) - (c2 + m(c1-c2)).(mp + \log(2k_2))}{c1.(mp + \log(2k_1))} \tag{14}$$

The energy savings in this stage can be calculated by Eq. (12), where D is in fact $D_{k2-anonymization}$.

Compression is a good additional mechanism to reduce energy by decreasing length of the messages during data transmission and reception. It can be applied to both anonymized and non-anonymized data. To show the relative benefit on energy consumption of our k-anonymity framework, we have not applied compression. However, WSN owners may also use compression at the outputs of our framework to further reduce energy consumption.

5. Performance evaluation

In this part, the trade-off between information loss and energy savings is investigated by applying k-ACM to synthetic data for different $k1$ and $k2$ values. A data record has five categorical attributes. Each attribute is considered to be a quasi-identifier having four distinct values. Synthetic data are generated randomly using the uniform distribution.

In Section 5.1, the performance of the $k1$-anonymization stage is evaluated. Information loss in the data shared with the semi trusted sink is especially an important evaluation criterion in this stage. Section 5.2 focuses on the performance of k-ACM in the $k2$-anonymization stage.

5.1. Performance evaluation of $k1$-anonymization stage

We analyze the performance of the $k1$-anonymization stage using information loss, anonymity level and energy saving metrics. In this way, the performance of the bottom-up clustering method as a general k-anonymity

Table 8. Experimental results of data set with 500 records for the $k1$-anonymous stage.

$k1$ value	Required Anonymity Level	Anonymity Level	Information Loss	Energy Saving
3	1.58	2.11	0.54	0.27
4	2	2.25	0.47	0.34
5	2.32	2.82	0.77	0.52
8	3	3.41	0.93	0.65

solution is investigated. Table 8 gives the performance values for a data set with 500 records. The column named 'Required Anonymity Level' gives the minimum anonymity score of the $k1$-anonymous data according to Eq. (9). This anonymity score is obtained if all clusters have exactly $k1$ elements at the end of k-ACM. However, the primary focus of our algorithm is making data at least $k1$-anonymous with minimum information loss. Therefore, the number of elements in the clusters may exceed $k1$. Due to this fact, the actual anonymity level of each case, which is shown in the column labelled 'Anonymity Level', is generally greater than the corresponding 'required anonymity level' value. The 'Information Loss' column gives the information loss of the $k1$-anonymity operation by using the conditional entropy of Eq. (4). The last column of Table 8, 'Energy Saving', gives the energy savings of k-ACM in the $k1$-anonymization stage within the energy consumption of the entire WSN due to the reformatting operation proposed in algorithm. In this reformatting operation, the iterated parts of the k-anonymous data are sent once along with the number of occurrences of these iterated parts in the data. In this way, the length of messages decreases. Energy savings at the end of the $k1$-anonymization stage is calculated using Eq. (12) and Eq. (13). In our analysis, we take the size of the WSN field as 500m x 500m, The size of each sub-region as 50m x 50m, and the transmission range, R, as $10m$. As expected, the information loss increases as the anonymity level increases. The information loss of 3-anonymous data is 0.54 and the whole system energy savings ratio is 0.27. 5-anonymous data provide an optimal solution, such that the information loss value, 0.77, is tolerable and the energy savings, 0.52, is quite high. For 8-anonymous data, the energy saving is very high (0.65), however, information loss is also high (0.93), making the data quality low.

In Figure 11, the effect of change in the number of records on the information loss is analyzed for various $k1$ values. For a given $k1$ value, the information loss in general decreases as the number of records increases. The main reason behind this decrease is that As the number of records increase, the data naturally becomes $k1$-anonymous and fewer generalizations are performed. However, in a few cases (e.g. transition from 500 to 600 records for $k1 = 8$), the information loss increases. Such exceptions are due to the nature of the clustering mechanism. The clustering mechanism may occasionally cause a higher average number of data records per cluster as the number of records increases. This, in turn, may cause a slight increase in information loss in some cases.

In the above experiments, we use data which are generated randomly using a uniform distribution. However, we also performed experiments using data which are generated by some non-uniform irregular distributions. In terms of information loss, the results for data with non-uniform distributions are better than the ones with uniform distribution in these experiments. If the data are produced by non-uniform distributions, some input vectors are closer to each other. Information loss for those vectors is lower at the end of the anonymization process. In other words, uniformly distributed data yields the worst case in terms of performance for our method in Table 8 and Figure 11. For simplicity and clarity, we do not present the experimental results obtained using non-uniformly distributed data.

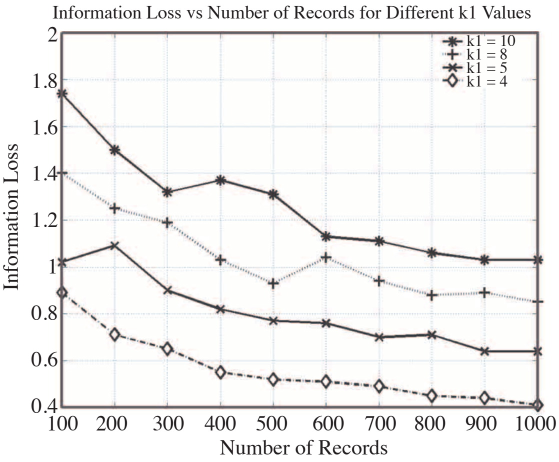


Figure 11. Information loss versus record number for different $k1$ values

5.2. Performance evaluation of $k2$-anonymization stage

The $k2$-anonymization stage starts with the $k1$-anonymous data obtained from the previous stage. The output enlargement factor, M, adjusts the length of the output that is obtained at the end of the $k2$-anonymization stage. As M increases, the size of the output, and, consequently, the number of encrypted data entries, increases as described in Section 4.3.2. This increase yields a more accurate output and decreases the amount of information loss.

Table 9, shows experimental results for a data set having 500 records, where $k1 = 4$ and $k2 = 16$. The first column lists the value of the output enlargement factor, M, that is pre-determined for the corresponding experiment. The second column shows the anonymity level that the output provides and is computed according to Eq. (10). This anonymity level is obtained after the encrypted parts are decrypted at the sink. The third column gives the information loss incurred in $k2$-anonymization stage alone, and is the difference between the information loss incurred at the end of the $k2$-anonymization stage and at the end of $k1$-anonymization stage. Eq. (4) is used to compute information losses. The energy savings of the $k2$-anonymization stage are given in the fourth column. The decrease ratio of this stage is calculated with Eq. (14) and the energy savings is found with Eq. (12). The energy saving indicates the Amount of energy saved in comparison to the extreme case where all $k1$-anonymous data are encrypted. The last row shows this extreme case. Here, for maximum output length, $M = 1$, and all the data are encrypted. Therefore, the information loss and energy savings are zero. The first row shows the other extreme case, in which the data are $k2$-anonymized using only generalizations without any encryption. The anonymity level of 4.88 indicates that the data received at the sink are already 16-anonymous, which is more than sufficient. In this case, the quality of data is very low, but the energy savings has a maximum value of 0.76. When M is increased to 0.25, some of the data entries are encrypted. The anonymity level of decrypted data at the sink decreases to 3.72 and the energy saving decreases to 0.54.

The information loss and energy saving trade-off can easily observed in Table 9. As the information loss decreases, the energy savings also decreases. A WSN designer can decide on the value of M according to the experimental results for information loss and energy savings, along with the system parameters and requirements. If the system can tolerate more information loss, it is possible to save considerable amount of

Table 9. Experimental results of data set with 500 records for the $k2$-anonymization part.

Output Enlargement Ratio	Anonymity Level	Information Loss	Energy Saving
0.0	4.88	0.95	0.76
0.25	3.72	0.54	0.57
0.50	3.07	0.29	0.38
0.75	2.67	0.13	0.19
1.00	2.25	0.00	0.00

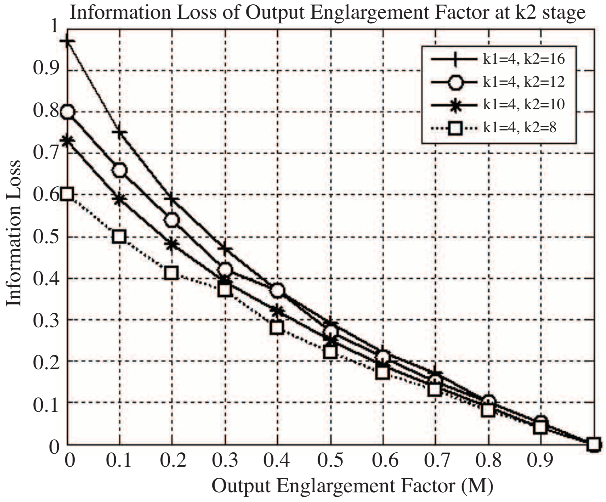


Figure 12. Output enlargement factor vs. information loss in the $k2$-anonymization stage.

energy. If energy consumption is not an important issue in the WSN, the quality of data can be easily increased.

The information loss at the $k2$-anonymization stage is analyzed in Figure 12 using different M values and different $k1$, $k2$ pairs for 500 records. As expected, an increase in M decreases the data loss.Moreover, high $k2$ values cause greater information loss in comparison to low $k2$ values. However, this difference becomes marginal as M increases. Figure 12 also shows that especially for high $k2$ values, k-ACM reduces the information loss quickly for small M values.

Figure 13 shows the variation of energy savings with respect to the output enlargement factor for various $(k1, k2)$ pairs. As expected, higher M values result in lower energy savings. For the information loss case shown in Figure 12, high $k2$ values yield more energy savings, but this advantage becomes marginal as M increases. However, contrary to the information loss decrease shown in Figure 12, the decrease in energy savings is linear.

The purpose of k-ACM is the maximization of energy savings and minimization of information loss. This tradeoff is managed via M. To do a better analysis of this tradeoff in the $k2$-anonymization stage, we analyze the ratio of energy savings to the information loss (ES/IL) for networks with different $k1$ and $k2$ values. These results are depicted in Figure 14. A higher ES/IL ratio shows the effectiveness of the output enlargement factor at this stage. It is observed that the ES/IL ratio constantly increases as M increases. This fact constitutes another verification of the effectiveness of the k-ACM algorithm. Although ES/IL has a constant increasing pattern, this does not mean that the highest (i.e. 1.0) M value must be chosen all the time, since such a high M does not help to save a considerable amount of energy, as shown in Figure 13. In fact, network administrators should choose the largest M value that yields the desired information loss and/or

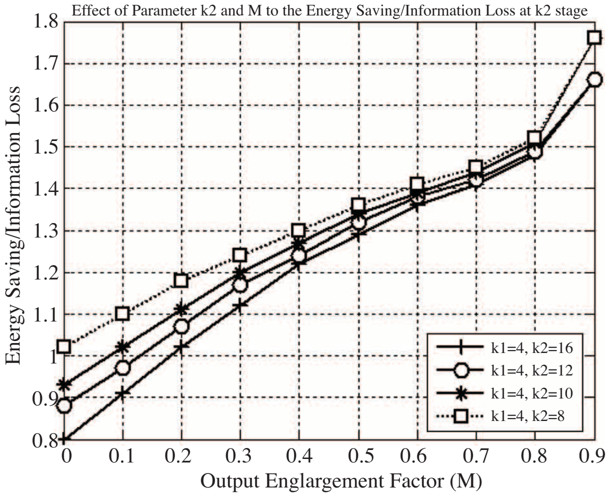


Figure 13. Output enlargement factor vs. energy savings in the $k2$-anonymization stage.

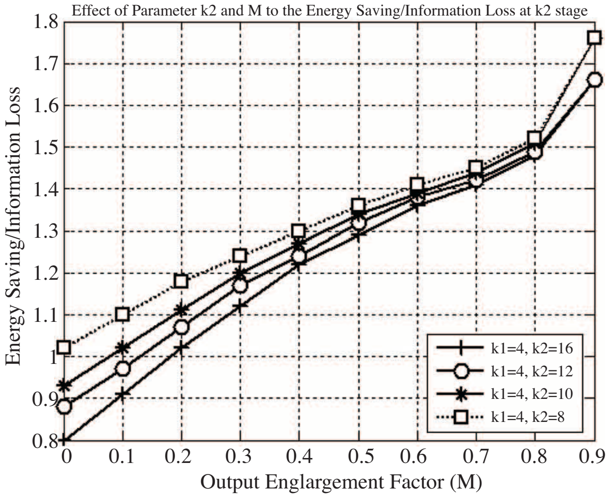


Figure 14. Energy saving/information loss for different M values at the $k2$-anonymization stage

energy saving values. In this way, the tradeoff between information loss and energy savings can be dynamically managed by playing with M. For example, consider a network with $k1 = 4$ and $k2 = 10$. If the maximum information loss this network can tolerate is 0.40, then Figure 12 suggests to use $M = 0.3$. In this situation, the network provides the required anonymity levels by saving 47% energy, as shown in Figure 13. However, if for the same network, the limitation is to save at least 60% of energy, then M is chosen as 0.10. In this case, the information loss becomes 0.59 for the same anonymity levels.

6. Related Work

Anonymity solutions in the literature are mainly based on two theoretical studies: DC-Nets and mixes [2, 3]. The basic idea behind DC-Nets is to anonymously broadcast a message to ensure receiver anonymity. If the message is intended to be sent to a specific destination, DC-Nets use secret and public key cryptography. However, DC-Nets only address the problem of recipient anonymity; sender anonymity is not considered. Performing secret key and public key distribution can also be difficult and inefficient in large networks . Mixes is an important idea for providing sender-recipient anonymity. In mixes solutions, nodes iteratively encrypt the message at each hop and relay the entire message to the next hop. No eavesdropper will be able to deduce the destination and source information from the content of the message. The mixes concept is used in practical applications, such as ISDN networks [4], e-mail and web applications [5, 6].

Mobile ad hoc networks benefit from anonymous routing protocols in providing sender-receiver anonymity [7, 8] by adapting Mix-net ideas. Since every node behaves as a router in these networks, the exchange of routing information comprises a considerably important amount of the legitimate traffic. Malicious users can deduce the sender and receiver of a message from the routing messages.

To the best of our knowledge, there are few studies about the anonymity problem in WSNs, most of which attempt to hide information about the location or time of events. Gruteser, et. al. [24, 25] proposed anonymity solutions for providing a high degree of privacy in sensor networks for location-based services, such as traffic monitoring, fleet management and 'pay as you drive' insurance. They claim that adversarial eavesdropping on event messages can identify a specific individual by linking the event location information with apriori knowledge about the event. In this study, however, the location and time information of events are cloaked so that such an outsider cannot differentiate any individual from among the other k different individuals.

Ozturk et al. [26] proposed a phantom routing method for hiding location information of originator sensor nodes in a sensor network geared towards tracking moving objects. The threat model is based on the existence of only one movable adversary node in the environment. The adversary tries to reach a moving target object by eavesdropping on the traffic and finding the previous hop of routing messages. The routing algorithm proposed aims to make such catching operations difficult by hiding the location of sensor nodes.

Privacy protection of the location of a receiver in a WSN is provided by a routing protocol in [27]. The proposed routing protocol prevents the eavesdropper from identifying the receiver by tracing the wireless packets. It randomizes the routing paths and injects fake packets in order to mislead eavesdroppers. Wadaa et al. [28] studied providing anonymity of the coordinate system, cluster and routing structures during the network setup of a WSN. Castelluccia et al. [29] proposed a homomorphic encryption method that securely aggregates sensor findings in an energy efficient way. This work deals with the aggregation functions, which compute the average or variance of sensor findings. Protection of location privacy is guaranteed by k-anonymity in location-based services that are given on mobile networks [30]. In this work, each mobile client specifies a minimum level of anonymity, and maximum temporal and spatial tolerances. The proposed methods try to provide the needed anonymity level within these quality of service parameters.

Privacy issues are becoming more and more important for data mining applications in which organizations share information with each other. Privacy preserving methods in data mining applications are explored in some database studies [31, 32]. These studies use perturbation methods on confidential parts of data. In this way, the adversary can obtain only perturbed data that has different distribution characteristics. Perturbation is done so that the application of data mining techniques to perturbed data will result the same accurate mining results. Generally, the organization who shares its data does not know the details of the analysis or data mining tasks which will be performed by data recipient party. This may be because the organization lacks technical expertise in the corresponding analysis methods or does not even know what type of analysis will be done on shared data by the data recipient. Therefore, privacy preserved data mining techniques cannot be used in these situations. Without detailed consideration of analysis methods, the data publisher shares information as much as possible, a situation which is known as privacy preserving data publishing [10].

Re-identification attacks identify individuals by linking information from various information sources [9]. To prevent this type of attack, this study presents a new anonymity approach, k-anonymity. In [33] and [34], k-anonymity is presented as a formal protection model. Sweeney [20], provides a formal presentation of combining generalization and suppression to achieve k-anonymity. This study uses generalization hierarchies during the generalization and suppression operations. Domain generalization hierarchies are introduced for categorical attributes and value generalization hierarchies are introduced for numeric attributes. Meyerson and Williams [11] showed that k-anonymization with a minimum number of suppressions is NP-hard. Aggarwal et al. showed that the problem of k-anonymization is NP-hard even when the attribute values are ternary [23]. Some approximation algorithms are proposed for this problem in [11] and [35]. Greedy heuristic algorithms are introduced in [19] and [21] to produce k-anonymous data while preserving the property of building decision tree classifiers. Therefore, privacy of data is guaranteed and can be used for classification purposes.

k-anonymity solutions solve the prevention of "record linkage attacks" in which the owner of a record is found through quasi-identifier attributes. However, it is shown that without finding the exact owner of a record, if a sensitive attribute exists in the record, such as the health status of a patient in a hospital database, it may be possible to identify the sensitive attribute of an individual under some circumstances by an "attribute linkage attack" [10]. k-Anonymity is extended by l-diversity, p-sensitivity and t-closeness notions to prevent such attribute linkage attacks [36, 37, 38]. Our study aims to prevent "record linkage attacks".

Studies about k-Anonymity and their extensions consider that the collected data is static. However, if the data is dynamic, in other words, if new records are inserted or existing records are deleted or modified as time passes, these solutions may show weaknesses. In applications that process dynamic data, anonymization has to be done in each of the many successive rounds. However, an attacker can deduce private information or find the owner of record by analyzing anonymization results of successive rounds. These anonymity solutions are extended to eliminate these weaknesses for dynamic sets in [39, 40, 41]. Byun et al. [39] studied data sets that are dynamically updated by record insertion . Xiao and Tao [40] extended this work by also including record deletion operations. Li and Zhou [41], studied record modifications with record deletion and insertion operations . WSNs collect dynamic data. In each collection period, record insertions are performed by sensor nodes. Our study does not address anonymity issues related to dynamic properties of WSN data collection. However, this challenging problem may be addressed by another dedicated study, building upon our proposed framework.

7. Conclusions

In this paper, we proposed a k-anonymization clustering method (k-ACM) that provides a k-anonymity framework for WSNs. Our threat model implies two levels of k-anonymity: 1) against the semi trusted sink, and 2) against eavesdroppers. Our threat model assumes the sink is a semi trusted entity, so that the data received by the sink must be at least $k1$-anonymous. To protect the data against eavesdroppers, data transmitted through the network must be at least $k2$-anonymous. Since the minimum protection against eavesdroppers must be greater than the minimum protection against the semi trusted sink, $k2$ is greater than $k1$. WSN designers can decide on the values of $k1$ and $k2$ by considering the security threats of the environment and application requirements. For example, military WSN applications can use higher $k1$ and $k2$ values during times of in war, and lower values during times of peace. If the possibility of an eavesdropper threat is high and the security of the sink is provided, then the WSN designers can choose higher $k2$ and lower $k1$ values.

In k-ACM, there is a tradeoff between data quality (in terms of information loss) received at the sink and energy consumption. The quality of data is reduced by generalization, since that operation irreversibly perturbs the data, thereby causing information loss.

k-ACM first makes the data $k1$-anonymous by applying generalization operations and continues to implement $k2$-anonymization via generalization and encryption operations. The output size of the $k2$-anonymization stage can be adjusted by a pre-determined threshold value, or *output enlargement factor.* As this ratio increases, the size of the output gets larger and more encryption operations are performed. During this enlargement, k-ACM selects the most suitable portions of the data for encryption to minimize information loss as much as possible. An increase of the output enlargement factor causes an increase in the energy consumption in the WSN. On the other hand, the quality of data received at the sink improves since the data is not greatly perturbed. If the ratio decreases, the quality of data also decreases, but the system consumes a considerably smaller amount of energy. In fact, k-ACM provides a mechanism for WSN designers to balance between information loss and energy cost by using the output enlargement factor. Our analyses show that the energy savings per information loss value constantly increases as the output enlargement factor increases. This implies that WSN designers should pick the maximum output enlargement factor that the information loss and/or energy saving restrictions of the WSN dictate. For example, our analysis shows that for a sink that is to receive 4-anonymous data (i.e. $k1 = 4$) and 12-anonymous data, which are required against eavesdroppers (i.e. $k2 = 12$), if the network can tolerate an information loss of entropy value 0.37, then WSN designers can pick an output enlargement factor of 0.4, resulting in an energy savings of 43% energy while also providing the required anonymity levels.

k-anonymization of data is performed by an algorithm that is based on UPGMA, a well-known bottom-up hierarchical clustering mechanism. The notion of conditional entropy from information theory is adapted for use as a distance function, which calculates the information loss during each clustering process. The same notion is also used for calculating the anonymity level of k-anonymous data and for evaluating the results of experiments in terms of information loss. Additionally, this paper introduces a dynamic taxonomy tree concept for the generalization operation.

References

[1] A.Pfitzmann, M. Khntopp. Anonymity, unobservability, and pseudonymity - A proposal for terminology. *in H. Federrath, editor, DIAU'00, Lecture Notes in Computer Science* 2009/2001: 1-9, 2000.

[2] D. Chaum. The dining cryptographers problem: Unconditional sender and receipent untraceability. *in Journal of Cryptology*, 1(1):65-75, 1988.

[3] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *in Communications of the Associations for Computing Machinery*, 24 (2):84-88, 1981.

[4] A. Pfitzmann, B. Pfitzmann, M. Waidner. ISDN-Mixes: Untraceable communication with very small bandwith overhead. *in GI/ITG Conference: Communication in Distributed Systems, Mannheim*, pp.451-463 Feb 1991.

[5] C. Gulcu, G. Tsudik. Mixing e-mail with BABEL. *in Symposium on Network and Distributed Systems Security (NDDS '96)*, San Diego,California, 1996.

[6] M. K. Reiter, A.D. Rubin. Anonymous web transactions with crowds. *in Communications of the ACM*, 42(2):32-48, Feb 1999.

[7] M. G. Reed, P.F. Syverson, D.M. Goldschlag. Anonymous connections and onion routing. *in IEEE Journal on Selected Areas in Communications,*16 (4): 482-494, May 1998.

[8] J. Kong, X. Hong, M. Gerla. An anonymous on demand routing protocol with untraceable routes for mobile ad hoc networks. *in Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 291-302, 2003.

[9] P. Samarati, L. Sweeney. Generalizing data to provide anonymity when disclosing information. *in Proc. of the 17th ACM SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems,* 1998.

[10] B. C. M. Fung, K. Wang, R. Chen, P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. *in ACM Computing Surveys*, 2009.

[11] A. Meyerson, R. Williams. On the complexity of optimal k-anonymity. *in Proc. of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems,* June 2004.

[12] J. Yick, B. Mukherjee, D. Ghosal. Wireless sensor network survey. *in Computer Networks*, Vol:52 No: 12 page: 2292-2330, 2008.

[13] D. W. Carman, P.S. Kruus, B. J. Matt. Constraints and approaches for distributed sensor network security. NAI Laboratories, Tech. Rep. 00-010, 2000.

[14] C. D. Michener, R. R. Sokal. A quantitative approach to a problem in classification. *in Evolution,* 11:130-162, 1957.

[15] H. Chan, A. Perrig, D. Song. Random key predistribution schemes for sensor networks. *in Proceedings of 2003 Symposium on Security and Privacy,* 2003.

[16] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *in ACM Transactions on Information and System Security (TISSEC),* 8(2):228-258 May 2005.

[17] W. Heinzelman, J. Kulik, H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *in Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, 1999.

[18] C. Intanagonwiwat, R. Govindan, D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *in the Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, 2000.

[19] B. C. M. Fung, K. Wang, P. S. Yu. Top-down specialization for information and privacy preservation. *in Proc. of the 21st Int'l Conf. on Data Engineering*, April 2005.

[20] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowledge-Based Systems*, 10(5):571- 588, 2002.

[21] K. Wang, P. Yu, S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. *in Proc. of the 4th IEEE International Conference on Data Mining*, November 2004.

[22] P. Andritsos, V. Tzerpos. Software clustering based on information loss minimization. *in Proc. of 10th Working Conference on Reverse Engineering(WCRE'03),* page:334, 2003.

[23] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigraphy, D. Thomas, A. Zhu. k-anonymity: Algorithms and hardness. Technical Report, Stanford University, 2004.

[24] M. Gruteser, D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. *in First International Conference on Mobile Systems, Applications, Services (MobiSYS)*, USENIX, 2003.

[25] M. Gruteser, G. Schelle, A. Jain, R. Han, D. Grundwald. Privacy-aware location sensor networks. *in Proc. 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS)*, 2003.

[26] C. Ozturk, Y. Zhang, W. Trappe. Source-location privacy in energy-constrained sensor network routing. *in Proc. of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp.88-93, 2004.

[27] Y. Jian, S. Chen, Z. Zhang, L. Zhang. Protecting receiver-location privacy in wireless sensor networks. *in Proc. of IEEE INFOCOM*, 2007.

[28] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, K. Jones. On providing anonymity in wireless sensor networks. *in Proc. of the Tenth International Conference on Parallel and Distributed Systems (ICPADS'04)*, 1521, 2004.

[29] C. Castelluccia, E. Mykletun, G. Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. *in the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2005, page: 109-117, July 2005.

[30] B. Gedik, L. Liu. Protecting location privacy with personalized k-Anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, Volume. 7, No. 1, January 2008.

[31] R. Agrawal, R. Srikant. Privacy preserving data mining. *in Proc. of the ACM SIG-MOD Conference on Management of Data*, pp.439-450, May 2000.

[32] D. Agrawal, C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. *in Proc. of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principal of Database Systems*, pp. 247- 255, May 2001.

[33] P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010- 1027, 2001.

[34] L. Sweeney. k-anonymity: A model for protecting privacy. *Int'l Journal on Uncertainty, Fuziness, and Knowledge-based Systems*, 10(5):557-570, 2002.

[35] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigraphy, D. Thomas, A. Zhu. Anonymizing tables. *in Proc. of the 10th Int'l Conference on Database Theory,* January 2005.

[36] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity. *in Proc. 22nd Intnl. Conf. Data Engg. (ICDE)*, page:24, 2006.

[37] T.M. Truta, V. Bindu. Privacy protection: p-sensitive k-anonymity property. *in Proc. of the Workshop on Privacy Data Management, In Conjunction with 22th IEEE International Conference of Data Engineering (ICDE)*, Atlanta, Georgia, 2006.

[38] N. Li, T. Li, S. Venkatasubramanian. t-Closeness: privacy beyond k-anonymity and l-diversity. CERIAS Tech. Report 2007-78, Purdue University, 2007.

[39] J. W. Byun, Y. Sohn, E. Bertino and N. Li. ”Secure anonymization for incremental datasets.” in SDM, pp. 48-63, 2006.

[40] X. Xiao and Y. Tao. ”m-variance: towards privacy preserving re-publication of dynamic datasets.” in SIGMOD, pp. 689-700, 2007.

[41] F. Li and S. Zhou. ”Challenging more updates: Towards anonymous re-publication of fully dynamic datasets” in arXiv.org:0806.4703v2, 2008.

Appendix

Analytical derivation of energy saving with k-ACM

Suppose the WSN field has a size of $X_{region}.Y_{region}$. Aggregation nodes are uniformly deployed in this area. The sink is located at the middle of the region, so the coordinates of the sink are $(X_{region}/2, Y_{region}/2)$. Assume that aggregation nodes divide the entire region into sub-regions, each having a size of $(X_{sregion}, Y_{sregion})$. Each aggregation node is located in the middle of the corresponding sub-region. The sensor nodes are also uniformly distributed in each sub-region. The expected distance of a sensor node to an aggregation node, $d_{sregion}$, in a sub-region is calculated as follows:

$$d_{sregion} = \int_{x=0}^{X_{sregion}} \int_{y=0}^{Y_{sregion}} \sqrt{(x-(X_{sregion}/2))^2 + (y-(Y_{sregion}/2))^2} f(x) f(y) dx dy \tag{15}$$

Here $f(x)$ and $f(y)$ are the probability distribution functions of the sensor coordinates. Since sensor nodes are uniformly distributed in the sub-region, they are chosen as $f(x) = 1/X_{sregion}$ and $f(y) = 1/Y_{sregion}$. The expected number of hops an event message travels from a sensor node to the aggregation node is:

$$h_{sregion} = d_{sregion}/R \tag{16}$$

From the sensor node to the aggregation node, an event message travels $h_{sregion}$ hops, which is calculated as follows:

$$d_{sregion} = \int_{x=0}^{X_{sregion}} \int_{y=0}^{Y_{sregion}} \frac{\sqrt{(x-(X_{sregion}/2))^2 + (y-(Y_{sregion}/2))^2}}{X_{sregion} Y_{sregion} R} dx dy \tag{17}$$

The number of hops between aggregation node and sink, h_{region}, is calculated as follows:

$$d_{region} = \int_{x=0}^{X_{region}} \int_{y=0}^{Y_{region}} \frac{\sqrt{(x-(X_{region}/2))^2 + (y-(Y_{region}/2))^2}}{X_{region} Y_{region} R} dx dy \tag{18}$$

Table 10. Energy consumption ratios.

Energy Consumption Ratios	Ratio Value
Transmission/Reception	1.5
Transmission/Encryption	2333.34
Encryption/Decryption	1

In WSNs, event information flows from sensor node to aggregation node, then to the sink. k-anonymization operations take place at aggregation nodes so the shortening effect helps to consume less energy while transferring event data from aggregation node to the sink. These operations consume additional energy for encryption and decryption operations, but the energy spent for encryption and decryption is quite small as compared to energy spent for transmission and reception. Energy consumption parameters are determined according to the experimental results presented in [13]. We assume that the data is processed in Sensoria's WINS NG RF subsystem with MIPS R400 processor, with an AES encryption algorithm. The transmission/reception, transmission/encryption and encryption/decryption energy consumption ratios for identical lengths of data are shown in Table 10. The transmission and reception rate is taken as 10 Kbps and power is 10mW. In all energy calculations, only event data processes are taken into consideration. Energy consumed for exchanging routing information or energy that is exhausted during the idle times of sensors are excluded from calculations in order to accurately calculate the energy consumption of the proposed method. Suppose that WSN generates event messages which are e bytes long and we assume that transmission energy T_T is 1.5 units (the actual unit is not so important since we eventually calculate energy saving as a ratio), reception energy T_R is 1 unit, encryption and decryption energy, T_E and T_D, are 4.29e-4 units.

The total consumed energy without k-anonymization is denoted as C_N. In this case, all event messages are sent to the aggregation node, but the aggregation node just relays them to the sink without any performing k-anonymization operation. At each hop, each event packet is transmitted and received once, so the energy consumed in one hop is $(T_T + T_R)e$, which is actually $2.5e$ units. The number of hops that each event message is transmitted is $h_{region} + h_{sregion}$. The energy consumption, C_N, can now be calculated as follows:

$$C_N = (h_{region} + h_{sregion})(T_T + T_R)e = (h_{region} + h_{sregion})2.5e \tag{19}$$

Total consumed energy in the case where k-anonymization is used is denoted by C_K. The length of an event message, which is transferred from a sensor node to an aggregation node, is assumed to be e bytes. However, this length is reduced to $(1 - D)e$ after the aggregation node due to the shortening affect of the k-anonymization. Here, D is the decrease ratio of the k-anonymization operation. Suppose that β bytes of the event message are encrypted. The energy consumption, C_K, can now be calculated as follows:

$$C_K = h_{sregion}e(T_T + T_R) + h_{region}(1 - D)e(T_T + T_R) + \beta(T_E + T_D) \tag{20}$$

The total energy savings C_G is calculated as follows:

$$C_G = 1 - \frac{C_K}{C_N} = 1 - \frac{2.5h_{sregion}e + 2.5h_{region}(1 - D)e + 8.58\ 10^{-4}\beta}{(h_{region} + h_{sregion})2.5e} \tag{21}$$

Complexity analysis of k-ACM

Suppose that k-ACM works on an input consisting of n event records and each record has m attributes. All of the m attributes are quasi-identifiers and each attribute has V distinct attribute values. The initialization phase mainly calculates the initial distance matrix and the run time for this part is $O(n^2.m.V)$. Initially, there are n clusters and at the end of the k2-anonymization phase, the minimum number of clusters is $n/k2$. Therefore, $n-n/k2$ cluster combination operations occur. Cluster combination consists of finding the minimum distance in the distance matrix, and reorganizing the matrix so that the distance values of the new cluster are added and the distance values of previous clusters are removed. If a binary heap structure is used for finding the minimum distance, formation of the initial heap structure with n^2 elements is $O(n^2)$. In a heap, finding the minimum operation is $O(1)$. However, removing distances of merged clusters from the heap and adding the distances of the new cluster to the heap requires $2n$ deletion and n addition operations, which cost $O(n\log(n))$. Reorganization of the distance matrix can be done in a time of $O(n.m.V)$ sequentially to maintaining the heap. As a result, the cost of each cluster combination operation is $O(nlogn{+}nmV)$. Recall that the maximum number of cluster combination operations is $n - n/k2$, thus, the algorithm reaches the end of the k2-anonymization phase in $O(n^2logn+n^2mV)$. Output enlargement for partial encryption and formation of k-ACM output takes $O(n)$ time. In total, k-ACM takes $O(n^2logn + n^2.2mV)$. In wireless sensor network applications, m and V generally have lower values so they can be assumed as a constant factor. Thus, the run time can be fine-tuned to $O(n^2logn)$.