Privacy Preserving Data Collection Framework For User Centric Network Applications

by

Hayretdin Bahşi

Submitted to the
Faculty of Engineering and Natural Sciences
in partial fulfillment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY

Sabancı University
February, 2010

Privacy Preserving Data Collection Framework For User Centric Network Applications

APPROVED BY

Assoc. Prof. Albert Levi ..... (Thesis Supervisor)

Assoc. Prof. Özgür Erçetin .....

Assoc. Prof. Erkay Savaş .....

Assist. Prof. Yücel Saygın .....

Assoc. Prof. Ercan Solak .....

DATE OF APPROVAL:

to my dear wife Birsen, my beloved daughters Nilgün and Neşe

ABSTRACT

Privacy Preserving Data Collection Framework For User Centric Network Applications

Advances in mobile and ubiquitous computing increased the number of user centric applications that comes into all aspects of our lives. This situation has started to threaten our privacy and created a huge demand for development of privacy-aware applications. Comprehensive privacy protection mechanisms have to take all phases of data processing into considerations including data collection from users, storage of data in central servers, and sharing them with third parties. However, privacy studies in the literature generally bring solutions for sharing of collected information with third parties.

In this thesis, a privacy preserving data collection framework is proposed for user centric network applications. Framework provides privacy of data en route to data collector(s). We propose a generic bottom-up clustering method that utilizes k-anonymity or l-diversity concepts during anonymization. Entropy based metrics for information loss and anonymity level are defined and used in performance evaluations. Framework is adapted for networks having different data collector parties with different privacy levels.

Our framework is applied for two types of data collection applications: (i) privacy preserving data collection in wireless sensor networks, (ii) preservation of organizational privacy during collection of intrusion detection logs from different organizations.

Traditional data utility vs. privacy trade-off has one more dimension in wireless sensor networks. This dimension is minimization of bandwidth or energy consumption due to the limitations of tiny sensor nodes. Our analyses show that the pro-

posed framework presents a suitable trade-off mechanism among energy consumption minimization, data utility and privacy preservation in wireless sensor network applications with one or multiple sinks.

It is also demonstrated that our framework brings effective solution for preserving organizational privacy during sharing of intrusion detection logs among organizations and central security monitoring entity.

Özet

Günümüz Kullanıcı Eksenli Ağ Uygulamaları için Kişisel Gizliliği Sağlayan Bilgi Toplama Anaçatısı

Mobil ve yaygın bilişimdeki ilerlemeler, hayatımızın her alanına giren kullanıcı bazlı uygulamaların sayısını artırmıştır. Bu durum kişisel gizliliğimizi tehdit etmekte ve kişisel gizliliğe duyarlı uygulamaların geliştirilmesi konusunda büyük bir talep oluşturmaktadır. Kapsayıcı kişisel gizliliği koruyucu mekanizmaların data işlemenin tüm fazlarını kullanıcılardan datanın toplanması, datanın merkezi sunucularda korunması ve üçüncül şahıslarla paylaşılmasını da kapsayacak şekilde göz önüne alması gerekmektedir. Bununla birlikte literatürdeki kişisel gizlilik çalışmaları çoğunlukla toplanmış bilginin üçüncül şahıslarla paylaşılması konusunda çözümler getirmiştir.

Bu tezde, kişisel gizliliği sağlanmış bir data toplama anaçatısı önerilmiştir. Önerilen anaçatı, kişisel gizliliği data toplayıcıya giderken sağlamaktadır. Anonimleştirme sırasında k-anonimlik ve l-farklılık konseptlerini kullanan genel bir aşağıdan yukarıya kümele metodu önerilmiştir. Performans değerlendirmelerinde, entropi bazlı bilgi kaybı ve anonimlik seviyesi belirleme metrikleri kullanılmıştır. Anaçatımız, birden fazla her biri farklı kişisel gizlilik seviyelerine sahip olan data toplayıcılarına sahip ağlar için de uyarlanmıştır.

Anaçatımız, iki çeşit data toplama uygulamasında denenmiştir: (i) kablosuz sensör ağlarında kişisel gizliliği sağlanmış data toplanması, (ii) farklı organizasyonlardan saldırı tespit kayıtlarının toplanması sırasında organizasyonel gizliliğin sağlanması.

Kablosuz sensör ağlarında geleneksel kişisel gizlilik & data yararlılığı ikilemine ek olarak bir boyut daha vardır. Bu boyut, küçük sensör düğümlerinin sınırlamaları nedeniyle band genişliği ve enerjinin minimize edilmesi gerekliliğidir. Analizlerimiz

göstermektedir ki önerilen anaçatı, bir ya da birden çok data toplama merkezi içeren kablosuz sensör ağlarında enerji tüketiminin minimize edilmesi, data yararlılığı ve kişisel gizliliğin sağlanması arasında uygun bir denge mekanizması oluşturmaktadır.

Anaçatımızın, organizasyonlarla merkezi güvenlik izleme birimi arasında organizasyonel gizliliği sağlayacak şekilde saldırı tespit kayıtlarının paylaşılması için etkin bir mekanizma oluşturduğu da gösterilmiştir.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are living in a data-centric world due to enormous improvement in information system technologies. Personal data is created, transmitted, processed and stored very easily. Our daily lives get easier and more safer because of this dynamic nature of data, thanks to developed information systems. We can watch our homes by using cameras and Internet even we are very far away from our homes. Patients can be tracked by health monitoring systems while they are outside of the hospital. Public and private watchdog agencies implementing many security surveillance applications in order to prevent or detect malicious activities in living areas. We can get information from traffic monitoring applications for choosing most convenient route in crowded cities. Smart grid applications collect details of electricity consumptions of homes in order to minimize energy consumption and lower the cost of electricity distribution infrastructure. The list of all these applications can be easily extended.

Rapid proliferation of mobile computing devices such as PDAs, and smartphones have made huge contributions to data-centric world in terms of data access. Improvements in sensor technology facilitated data collection enormously. Data communications have been facilitated with the advent of wireless technologies, like WiFi, ad hoc or mesh networks. Database system gets more scalable and robust. Data storage and data processing capabilities have been efficiently improved. All these technological improvements enable us to deal with huge amount of data mostly over the environment which is generally un-controlled.

These benefits of advancements in information technologies lead to a very major problem: violation of privacy. Privacy is described as ability of an individual or group to decide which information about themselves would seclude or which information would revealed to whom. However, in such a data-centric world, information owners, in most of the time, cannot control flow of their personal data. Controlling requires putting some restrictions on data during creation, transmission and storage stages of data life-cycle. These restrictions are applied by privacy preservation mechanisms embedded in information systems. Preservation mechanisms are generally motivated and required by laws or other regulations like HIPAA (Health Insurance Portability and Accountability Act) [1]. Since individuals cannot protect their privacy by themselves, governments try to help individuals by regulations. Many countries promote and protect individual privacy by different acts.

Technically, privacy preservation mechanisms have to solve critical problems. First problem is privacy needs of individuals have to be determined according to well defined privacy criteria. These criteria have to be quantified by measurable privacy metrics. Second problem is that privacy preservation mechanisms have to solve trade-off between data utility and privacy. Privacy preservation imposes restrictions on data itself or mechanisms restrict creation, transmitting and storage of personal data, which means data utility is degraded to some extent. Preservation methods have to discriminate personal data from other data, fulfill all privacy requirements and, at the same time, they have to maximize data utility.

First responsibility of a privacy preservation mechanism is to collect data from individuals according to required privacy criteria. The second responsibility is to protect personal data against security threats of environment. This responsibility may include using appropriate encryption and access control mechanisms, performing procedural security countermeasures, providing physical security of environment, hardening information systems against remote and local hacking activities. In this thesis, we concentrate on privacy preserving data collection mechanisms.

Privacy Preserving Data Collection Models:

Studies on privacy problem mostly concentrated on achieving collection and sharing of data under the required privacy constraints in order to make efficient knowledge-based decisions. Data collection refers to conveying of user data from data owners to a central data collector. In some situations, data collector may share this data with other third parties. Data collection is modelled according to the trust level between data owner and data collector party. These models are categorized into two categories, *trusted data collection model* [2] and un-trusted data collection model [3].

In trusted data collection model, [2] as shown in Figure 1.1, data is collected from data owners by data collector party named as *data publisher*. Data publisher shares data with *data recipient* who will actually use it for performing a required data analysis task. Here, data collection and data sharing are separate operations. A generic example is the application where hospitals share medical records with medical research institutions. In this example, data owners are patients, data publishers are hospitals and data recipients are medical research institutions. Data publishers collect all the details of records of data owners $(R_1, R_2, \cdots R_n)$ but they are required to share privacy preserved data with data recipients. Data owners do not trust data recipient parties; however, they are required to fully trust data publishers in completing privacy preserving operations. Also data owners have to be sure that their data is not maliciously or unintentionally used for illegal duties by staff of data publishers.

An intrinsic assumption of this model is that data publisher does not know the details of analysis or data mining tasks which will be performed by data recipient. This may be due to situation that data publisher has lack of technical expertise in the corresponding analysis methods or data publisher does not even know what type of analysis will be done on shared data in data recipient part. According to this model, without detailed considerations about data analysis methods, data publisher shares information with data recipient as much as possible. However, criteria which

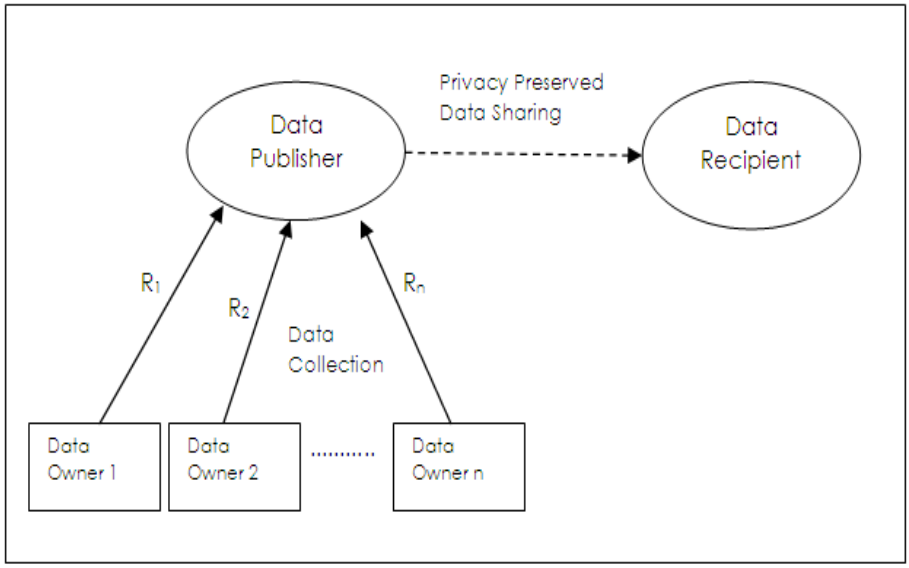


Figure 1.1: Trusted Data Collection Model

will fulfill the privacy needs of data owners have to be devised and they have to be obeyed during data sharing.

In un-trusted data collection model as shown in Figure 1.2, data owners send their records to data collector but they do not trust it. Data collection and data sharing do not occur separately as in trusted model. Privacy preservation has to be done at the data owner side and privacy preserved data $(R_1^{'}, R_2^{'}, \cdots R_n^{'})$ are collected by data collectors. Data is perturbed so that data collector cannot deduce records of individuals but the same data analysis results are reached by data collector. In this model, queries or data analysis methods those are used by data collector have to be known in advance. Therefore, these solutions are required to fix and restrict the type of analysis or mining tasks at the data collector side which may not be possible in most of the time.

Privacy Preservation Techniques

As a privacy preserving operation, data collector can mainly use two main techniques: (i) privacy preserved data publishing [2], (ii) privacy preserved data mining

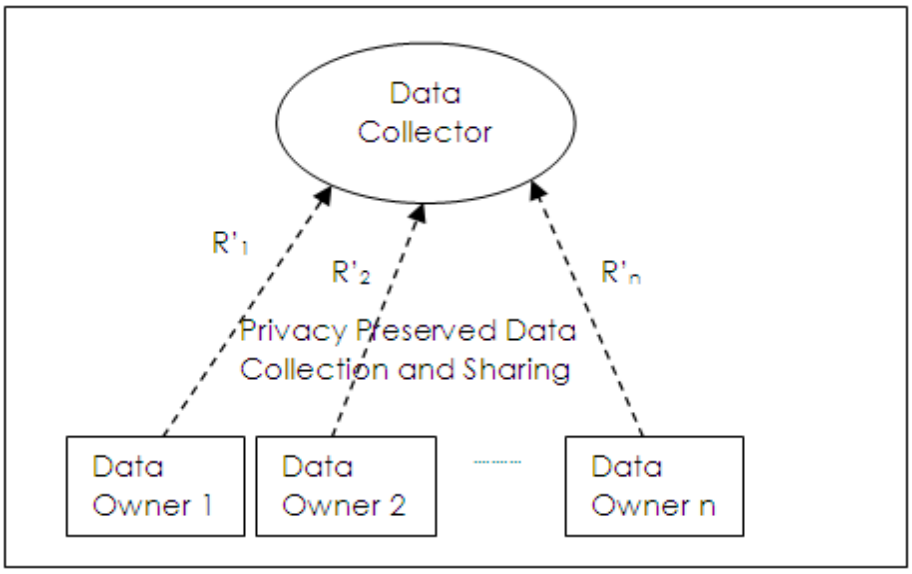


Figure 1.2: Un-trusted Data Collection Model

techniques [4]. Privacy preserved data mining techniques perturb the original data so that application of these techniques to perturbed data will result the same accurate mining results. However, data collector have no possibility to obtain original data which may violate the privacy of data owners. Some attributes of data records may be modified with false data, the attribute values of may be swapped among different records or totally new artificially created records can be inserted to original data sets at the data owner side. Un-trusted data collection model uses privacy preserved data mining techniques. Also in trusted data collection model, data publishers can benefit from these techniques before sharing the data with data recipients. However in practice data publishers do not know the details of data analysis tasks which will be performed by data recipients. For example, Californian hospitals have to publish patient records on the Web due to regulations [5] without exactly knowing the analysis types.

Privacy preserved data publishing techniques do not change the truthfulness of data at record level and try to collect information as much as possible. These techniques assert privacy criteria which can be applied to a collection of records. There-

fore, they need to collect a set of records at a trusted party for privacy preservation operations. Trusted data collection model applies these techniques. *k-Anonymity* is introduced as a basic privacy method in privacy preserved data publishing [6]. This method is based on the fact that privacy problem cannot easily be solved by just stripping of identity information (name, surname, social security number etc) from the record of data owner. Some other data fields called quasi-identifiers may be used to identify a person by using external information sources. This attack technique is called"Re-identification attac" [6] or "record linkage attack" [2]. For example in a hospital database, address, sex or other attributes can identify exactly a person. k-Anonymity generalizes or suppresses quasi-identifiers of data records so that any individual cannot be differentiated between other records of $k-1$ individuals by using those quasi-identifiers. k-anonymity solutions solve the prevention of "record linkage attack" which is actually finding the owner of a record through quasi-identifier attributes. However, it is shown that without finding the exact owner of a record, if sensitive attribute exists in a record (like health status of a patient), it may be possible to identify sensitive attribute of an individual in some circumstances by an attack called "attribute linkage attack" [2]. k-Anonymity is extended by l-diversity, p-sensitivity and t-closeness notions in order to prevent attribute linkage attack [7–9].

1.1 Motivation

Existing studies approach to the privacy problem in the context of the discussed trusted and un-trusted models. There is an intrinsic mapping between privacy models and privacy methods so that trusted model uses privacy preserved data publishing methods whereas un-trusted model uses privacy preserved data mining methods.

In network applications, which collect data from users, determination of analysis and mining tasks in the network design stage may limit the capability of this data collection system. Also, the requirements of analysis tasks may change with time.

Therefore, in most of the time privacy preserved data publishing methods are more useful in data collection applications. However, data owners may not fully trust data publisher or data collector party. Any person with malicious intent in the data collector site may have possibility to reach all the private data. Another possibility may be that due to inefficiency of security countermeasures at the data collector site or security problems during data collection operations, attackers can obtain private data. Data owners generally may want data to be privacy preserved before reaching these un-trusted parties. Therefore, a new model has to be devised so that privacy preserved data publishing methods can be applied for un-trusted data collectors. This thesis provides a model for application of publishing methods in the environments having these types of data collectors.

It seems that direct solution is performing privacy preserving operations directly at owner side by stripping off the identity information. However, record linkage or attribute linkage attacks threaten the privacy of data owners. Privacy preserved data publishing methods, k-anonymity, l-diversity, etc., prevent against these attacks but they can be applied where many data of users are collected. They can take place at somewhere between owner side and data collection party. This place has to be trusted by the data owners. If this place is chosen as close as possible to user and privacy preservation is done by automatic applications where any involvement of a person is not possible, users will be more satisfied about the privacy preserving mechanism. This criterion also leads system designers to set-up distributed trusted parties, which means if one trusted party is compromised by attackers, only some part of the data owners will be affected by this compromise. "Single point of failure" property of traditional trusted model will change the notion of distributed trusted parties. A required model for privacy preserved data collection is shown in Figure 1.3. In this model, data owners send their data to local trusted parties. Appropriate privacy preserving operations are done at these local ones and privacy preserved data is sent to the data collector, which is un-trusted for users. For example, if the data collection environment is mobile phone infrastructure, privacy preserving mechanisms can take place at the local base stations instead of performing them at

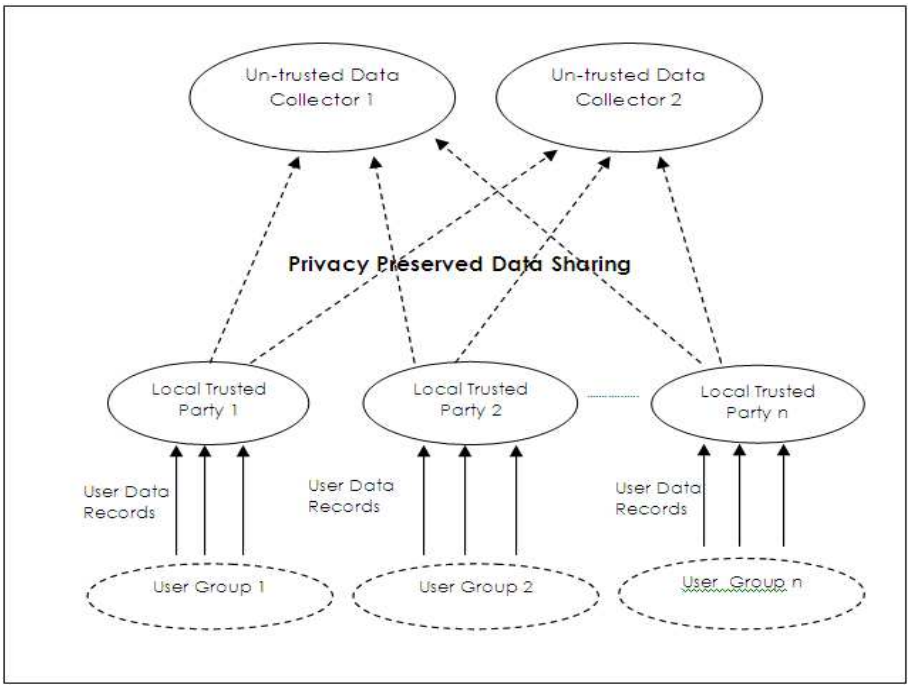


Figure 1.3: Distributed Trusted Data Collection Model

the central data collection point. Sometimes, designer of collection systems can also have options to find trusted parties between data owners and data collection centers. Suppose that a research center collects intrusion logs of organizations over Internet in order to do intrusion detection research but organizational privacy have to be also preserved during this collection. Organizations mostly do confidentiality agreements with their Internet Service Providers (ISPs), therefore ISPs may be trusted parties for organizations.

Privacy models assume that there is only one collection party in a data collection application. However, in the existing network applications, applications themselves may send information to two different collection parties at the same time. For example, health-care monitoring applications may send health status of an individual to hospital or to a relative of that individual at the same time. On the other side, due to nature of application, information can be captured by third parties such that an eavesdropper can gather wireless packets in a wireless sensor network (WSN) appli-

cation. Different data collectors may have different trust levels. A patient can have a more trust his relative than hospital or eavesdropper is totally an un-trusted party in a WSN application. Privacy preserving model and methods have to take all the receiving parties and their different trust levels into consideration. The framework proposed in this thesis provides a privacy model with one or more data receiving parties having different levels of privacy.

1.2 Challenges

Privacy preserving operations have to solve a trade-off between data utility and privacy level of data. Privacy preservation operations remove or modify portions of data in order to fit to privacy criteria. Privacy preserving methods have to cause data loss not more than the requirement of criteria but this is not an easy task. For k-anonymity, Meyerson and Williams [10] showed that k-anonymization with 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 [11]. Also decisions for the required privacy level have to be done according to the possible loss in the data utility and tolerance level of application to this amount of loss.

Requirement of privacy levels may not be fixed during the life-cycle of personal data. For example, applications like patient monitoring outside hospitals, continuously track spatio-temporal information of patients with their health statuses. In non-emergency times, users may not want to give full details of time and location knowledge to hospitals; however, in urgent times, users prefer to maximize the data utility so that immediate actions are performed by hospitals. Therefore, privacy levels imposed by applications have to be managed easily.

Most of the network applications have resource constraints so that privacy preservation operations have to take them into account. For example, wireless sensor network applications have to minimize energy consumption or wireless mesh networks have to reduce network bandwith usage at mesh routers and decrease energy

consumption at client side.

Privacy preserving data publishing methods need trusted parties. Although these methods cannot avoid the requirement of trusted party existence, additional privacy and security mechanisms may be developed for the transmission of data from users to trusted parties and storage of data at trusted party.

1.3 Design Objectives of the Proposed Framework

Proposed privacy preserved data collection framework bases on the property that data collection party collects as much information as possible in order to have wider coverage of analysis types. Therefore, our framework brings solution for privacy preserving data publishing methods like k-anonymity or l-diversity which are aiming to prevent record linkage or attribute linkage attacks.

Our framework assumes that data collector is not fully trusted party for users. Also it is designed so that more than one party having different levels of privacy can collect data at the same time. Privacy model has to be adopted so that privacy preserving publishing methods will work under these assumptions.

Minimization of energy or minimization of bandwith usage has to be considered as a major design criteria according to the type of network. Privacy preservation causes data loss. If preservation operations are done at closer places to users, this lost data does not need to be transferred unnecessarily in the network which may increase the resource consumptions. Representation of collected data has to be chosen such that size of collected data is minimized.

Privacy preservation methods have to maximize the data utility under the required privacy criteria. Generalization or suppression operations have to be cleverly done so that data collector party can reach more accurate analysis results. These operations have to be well adapted to the chosen representation types for collected data.

1.4 Contribution

In this thesis, a privacy preserving framework is proposed for applications in which data is collected from users. A privacy model for data collection is adapted so that there is no a central trusted data collector party; all privacy preserving operations are done in distributed trusted parties which are located closer to users. Moreover, our model assumes that there exists more than one data collector party having different privacy levels.

Bottom-up clustering idea is adopted as an anonymization method in order to have multiple privacy levels. Traditionally, suppression and generalization operations are used for anonymization. Proposed framework uses encryption instead of suppression for achieving multiple levels of privacy in one anonymous output for different data collectors. Encrypted data portions can be recovered by the parties having corresponding keys. Parties having lack of appropriate keys consider these data portions as suppressed data.

Proposed framework minimizes the data loss of generalization operations by introducing the dynamic taxonomy tree concept. If two different values are generalized according to static taxonomy trees, attributes are replaced with an attribute in the common ancestors of these two values. Instead, dynamic taxonomy tree method sends a set of attribute values instead of more generalized value in order to create more quality data. However, this may increase the data length in some situations. Analysis of information loss versus resource consumption trade-off is done in related parts.

Proposed privacy model and framework are deeply investigated in two different network applications. Firstly, they are adopted for wireless sensor networks (WSNs). For different types of WSN topologies, energy consumption, information loss and multiple levels of privacy issues are explored under the proposed model and framework. Secondly, adaptation performed for providing of organizational privacy in an application that collects intrusion logs from organizations over the Internet in order to perform collaborative intrusion detection research.

1.5 Thesis Outline

Outline of this thesis is organized as follows: Chapter 2 gives background information about privacy preservation concepts and presents related work in the literature. Chapter 3 introduces the proposed anonymization method, k-Anonymization Clustering Method. Chapter 4 proposes a k-anonymity based framework for privacy preserving data collection in WSNs. In Chapter 5, privacy framework is proposed for preserving organizational privacy in log sharing applications. Chapter 7 concludes the thesis.

Chapter 2

Background Information and Related Work

Privacy has been studied under the name of "anonymity" for a long time. Anonymity is defined as subject being not identifiable within a set of subjects [12]. Therefore, hiding of a subject among other subjects may be a well-defined privacy criteria. Privacy and anonymity terms are generally used instead of each other. However, privacy has a more comprehensive meaning.

Privacy or anonymity has been studied in two different information technology fields, *database* and *network*. General road map of related studies differ in both areas. In network studies, anonymity problem is mostly referred to hiding sender and/or receiver identities of network messages. On the other side, database community has a data-centric approach which targets to hide the owner of a data record or hide the disclosure of sensitive attributes of a record owner. Database community started with the notion of k-Anonymity and later has extended it.

In this section, first two subsections, 2.1 and 2.2, give background information and literature review about sender/receiver anonymity and k-Anonymity, respectively. Subsection 2.3 gives some information about entropy notion as a privacy and information loss metric. Subsection 2.4 specifically concentrates on privacy studies about wireless sensor networks. Subsection 2.5 reviews the studies about the privacy of intrusion and security logs.

2.1 Sender/Receiver Anonymity

Subjects of anonymity problem are generally chosen as the identities of communicated nodes in various types of networks. Possible sender or receiver nodes constitute the anonymity set. In anonymous system, an outsider could not discriminate the communicated parties among the members of anonymity set. Therefore, larger anonymity sets mean more anonymous systems.

An adversary who monitors and/or captures certain parts of the communication system may want to know the receiver of a message, the sender of a message or matching pair of sender and receiver in critical communications [12]. Receiver anonymity prevents the identification of message receiver, sender anonymity prevents finding of message sender and relationship anonymity prohibits the determination of relationship between sender-receiver pairs. Hiding the relationship is termed as un-linkability of sender and receiver [13].

In normal network communications, source and destination of messages are embedded in packet headers. Although elimination of this information can be somehow achieved, various traffic analysis methods [14] can be applied by attackers in order to obtain discriminative communication patterns and using them for the identification of communicating partners. Patterns can be deduced from time and duration of communication or the length of exchanged data. Providing confidentiality or integrity for exchanged data cannot prevent the traffic analysis. Real anonymity can be achieved by providing *unobservability* property. *Unobservability* is described as the state of communication message being indistinguishable from any other messages at all [13]. This property ensures hiding of all communication patterns.

Sender/receiver anonymity problem in the literature are mainly based on two theoretic studies; *DC-Nets* and *mixes*.

The basic idea of DC-Nets [15] is anonymously broadcasting a message for providing receiver anonymity. If the message is intended to send to a specific destination, it is encrypted by the destination's public key. In addition to public keys, each node shares a secret key with other participants and use them in anonymously sharing

data. DC-Nets suffer from many important drawbacks. First of all, it brings solution for only recipient anonymity; sender anonymity is not considered. Second, performing secret key and public key distribution could be difficult in large networks. Third, if two participants send messages at the same time, two messages will be added at each participant where any participant including actual receiver cannot receive the message. Therefore, sending one message at a time must be guaranteed by the system. This drawback can lead to a very effective DoS attack such that malicious users constantly send messages in order to deliberately spoil the content of the actual delivered message.

Mixes have got higher attraction among the researchers so that many papers are published and practical applications are created by using the various versions of the mix idea. Mixes [16] is an important idea for providing sender-recipient anonymity and resisting to traffic analysis. Let's say a sender sends a message to a recipient through the forwarding nodes $f_1, f_2, f_3, ..., f_D$ and all of these nodes are known by the sender. Firstly sender encrypts the message with the public key of f_D, and then consequently with the keys of $f_{D-1}, f_{D-2}, ..., f_2, f_1$. Suppose that public key encryption is denoted by $E_{f_x}(M)$ where M is the plain text and f_x is the public key. So sequential encryption yields the following structure: $E_{f_D}(E_{f_{D-1}}(E_{f_{D-2}}(..., E_{f_2}(E_{f_1}(M)))))$

Each node on the message path, gets the message, decrypts it with its private key and sends to the next hop. The first node of the path knows the actual sender and the last node knows the actual receiver. Therefore, the remaining nodes only have information that they received a message from the previous node and relayed it to the next hop. Any eavesdropping activity cannot deduce the destination and source information from the content of the message; however, traffic analysis may yield valuable information. In order to prevent these attacks, each intermediate node does not immediately deliver the messages to the next hops and stores until a predetermined number of other messages are arrived to this node. Then, the node delivers the messages in random order so that any eavesdroppers cannot correlate the incoming and outgoing messages and trace the full path. This schema does not only supply resistance to passive eavesdropping attacks but also it can prevent

active attacks. Since each forwarding node knows only the previous and the next hop, attacker does not have the ability of tracing the message unless he captures all nodes of the path.

Although this schema promises higher security assurance level for passive and active attacks, it suffers from various practicality issues. First of all, it introduces additional latency in each forwarding node. There are different approaches for message delivery methods which try to balance the trade-off between latency and resistance to traffic analysis [17, 18]. However, latency remains as an important problem in acceptably secure systems.

Anonymity problem must be handled in different network types ranging from the Internet to ad hoc networks. There are various motivations of this problem in different networks which are described below.

In the Internet, users may not want to reveal their surfing habits like the sites those they visit, time intervals or duration of their site visitings. Another possibility is that companies sharing information over Internet may not want to reveal whether or not any sharing operation has been done with each other.

Web proxies are used in contemporary systems for accelerating the web access to the Internet. They also provide sender anonymity property to a limited extent. All users in the internal network constitute the sender anonymity set. Firewalls and routers also provide similar functionality by *network address translation* (NAT) technology. Anonymizer [19] uses a similar proxy model but it is created for providing anonymity, not for accelerating the web access.

Other than using one proxy, systems called *Onion Routing* [20] use a series of proxies which cooperate for sending the web requests. This system is based on the idea of mixes so that the source chooses the path and the message is multiplicatively encrypted. Each forwarding node decrypts one layer, gets the next hop information and relays it. Because of the real time requirements of the applications, onion routers differ from mix-nets in delaying and reordering the traffic at each node. However, onion routers can send different types of traffic to each other over a single channel; therefore, traffic analysis cannot practically be helpful to the attackers in busy onion

networks.

Crowds [21] is another practical anonymization system. The users of the system form Crowds users group and the connection request to a web server outside of this group is directed to a random user inside the group. The new node has two possibilities: (i) either passing the request to the web server, or (ii) passing it to another randomly chosen node of crowds group. The request travels within the group until one of them chooses to send it to the destination web server. Each transmission is encrypted with the destination's shared key. Local eavesdroppers, who can observe all local transmissions in the group, cannot determine the receiver unless they capture any node along the path or use any traffic analysis methods. On the other side, the sender of the request is not the actual sender. Thus sender anonymity is only provided for the eavesdroppers, who can monitor from the outside of group. This system does not solve the global eavesdropper problems and it does not take traffic analysis issues into account. Another study, named *Hordes* [22], uses the same idea with Crowds except it additionally uses multicast service to anonymously forward the replies to the sender.

In highly mobile ad hoc networks, new problems may arise like the need for hinderance of location information and motion patterns of the nodes from malicious users. Especially in critical military applications, obtaining motion patterns and locations of mobile clients may be very important for the enemies [23]. In such a network, motion pattern inference is very easy because of the excessive routing messages. These messages reveal the location information since node determines the next hop from its RF coverage area and finding the path of a message makes an adversary predict the relative distances between each hop. By combining the path information with the location knowledge of the eavesdropping nodes, near-exact node positions are found out. Enemy can infer the motion patterns by periodic examinations of the node locations. Deducing the message path is not a very big deal in mobile ad hoc networks because of their high routing properties. There are not any dedicated hosts for routing operation; routing messages are routed by every host that exists on the message path. In the most common ad hoc routing

protocols, routing messages are sent in clear text format so that any malicious node can listen and deduce the full message path. In *Dynamic Source Routing* (DSR) [24] protocol, all path information is stored in the message. Source, destination and all other forwarding nodes can be deduced from a single intercepted message. In *Ad hoc On-Demand Distance Vector Routing Protocol* (AODV) [25], routing information is stored in routing tables and it is exchanged whenever needed. Although finding out of the trace is not trivial as in DSR, this routing method does not totally solve the problem, because collaborative eavesdroppers can trace the message from source to destination and learns the message path. Therefore on demand protocols like AODV fail to achieve location and motion privacy under the assumption that enemy has unbounded eavesdropping capabilities.

Mobile ad hoc networks can benefit from anonymity solutions in providing mobile privacy. Since every node behaves as a router in these networks, exchange of routing information occupies considerably important amount of the legitimate traffic. Malicious users can use routing messages for their eavesdropping aims and they can easily violate mobile privacy and anonymity of the system. First of all, anonymous routing protocols must be developed. Then the messages must be sent in this routing structure anonymously.

ANODR (anonymous on demand routing) [23] protocol is proposed for anonymous routing. Tracing an on-demand routing protocol is more complicated than tracing other ones, but collaborative eavesdroppers can easily track a communication. ANODR anonymously discovers the route by using the notion of broadcasting with trapdoor information, which is based on onion routing idea of Mix-net. In onion routing system, source encrypts the message sequentially by all of the public keys of nodes on the path starting from the actual destination node to the first node of path. It is important to note that full onion message is created in the source node and it is assumed that the source knows the full path. In starting phase of ANODR protocol, since it is on demand protocol, source does not know the path, protocol tries to find it by route discovery process. First, source broadcasts route discovery message to the nodes in its coverage area with a trapdoor function embedded in the

message. Each node that takes the broadcast message determines if the message is destined to itself by using this trapdoor function. If a node understands that the destination is not itself, it encrypts the entire message with its public key and broadcasts again. Each broadcast operation adds one layer to onion at each node until the message reaches to the destination. Destination reverses the onion to the sender after determining that it is destined to itself. After that each hop decrypts the message by its private key, puts appropriate route pseudonyms and relays the message to the next hop. After all route pseudonyms are constructed for full path, route discovery phase is completed and the actual message is ready for sending operation. Receiver anonymity is provided by broadcasting and content correlation is prevented by onion structure and route pseudonyms.

Privacy studies of network community generally focus on hiding sender or receiver entities in network communications. However, in data collection applications, sender or receiver entities are known by all parties. Privacy threat models of data collection applications concentrate on privacy of collected data rather than hiding the communicated entities. In this thesis, collected data is considered as the subject of privacy.

2.2 Basics of k-Anonymity

At first glance, it may be assumed that privacy problem can be easily solved by stripping off the attributes which identify individuals like name, social security number etc. However, the problem of privacy poses extra challenges that cannot be easily solved by simple "stripping off" mechanisms. Some other data fields and sources may be jointly used to deduce some private information. Suppose some organizations need to share their electronic information, such as public health or demographic data, with other organizations. However, they want to provide the privacy of their consumers or personnel during this information sharing operation. Simply stripping off the name or social security number information from the data set does not solve the privacy problem. It is possible to identify the owner of a record by using

attributes like birth date, address, sex, ZIP code etc. Government or public organizations release data in the form of voter lists, telephone or address books, local census data. Collecting data from these resources and linking them to targeted released data may enable to identify the individuals. Also some people can collect data directly observing specific individuals and use their observations and released data in order to obtain private information of individuals.

Attack method called 're-identification attack' [6] or 'record linkage attack' [2] directly uses these methods. In order to prevent these type of attacks, Samarati and Sweeney proposed k-anonymity [6]. Basically, k-anonymity brings a specific restriction to anonymity problem so that it targets to hide one subject among some other $k-1$ subjects. In other words, the attributes those 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, 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. Generalization and suppression operations 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 [10].

2.2.1 k-Anonymity Definitions

Some basic definitions used used in k-anonymity are explained below:

Quasi-identifier Attribute: Attribute that is not able to identify a subject by using per se but it may help to identify subject with the 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 in table T. T has 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 to make the data k-anonymous.

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

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

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

2.2.2 Taxonomy Trees

Taxonomy tree is a tree structure that is created for each categorical quasi-identifier attribute to replace existing attribute value with more general one in k-anonymization process [26–28]. Actually, this replacement is a generalization operation. Leaves of the tree contain the distinct values of attributes. Nodes in the higher levels of 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, that means the attribute value has no information. Suppression is considered as another operation for anonymization in literature [27], but it is actually a generalization operation where the attribute value is generalized to the root of attribute's taxonomy tree.

Let us think that a sensor network collects location information as an address. A possible taxonomy tree for this location attribute can be constructed as in Figure 2.1. Suppose that k is chosen as 2 and location attribute is the quasi-identifier. If there are two records having location information for 'Buket Street' and 'Selvi Street', and if they are decided to be anonymized to a common value, location attribute value is generalized to common ancestor in the tree which is actually 'Istasyon Avenue'. Location attribute values of two records are replaced with this

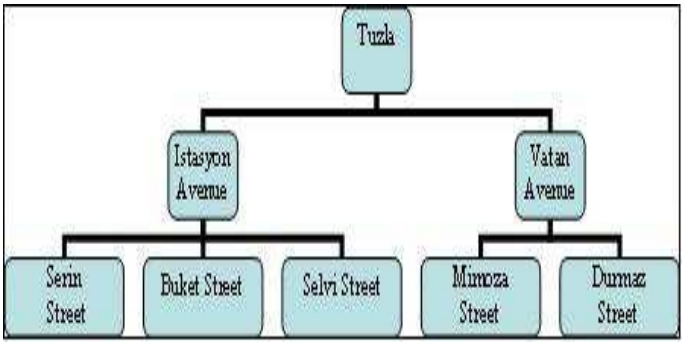


Figure 2.1: Static Taxonomy Tree for Location Information

Table 2.1: 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 Str., Buket Str., Selvi Str., Mimoza Str., Durmaz Str.

more general attribute value. Thus, no one can discriminate these two records from each other by using location information.

2.2.3 $k-$Anonymity Example

For example, assume that a wireless sensor network is constructed for traffic monitoring. This application collects information about the vehicles passing through some locations of a city. Attributes of sample data is given in Table 2.1. A sample set of data is given in Table 2.2. k is chosen as two and data in Table 2.2 is made 2-anonymous. A sample 2-anonymized version of the data by only generalization operations is shown in Table 2.3. Anonymization operations are completed by using taxonomy trees for location information and vehicle type which are given in Figure 2.1 and Figure 2.2 respectively.

Table 2.2: A 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

Table 2.3: 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

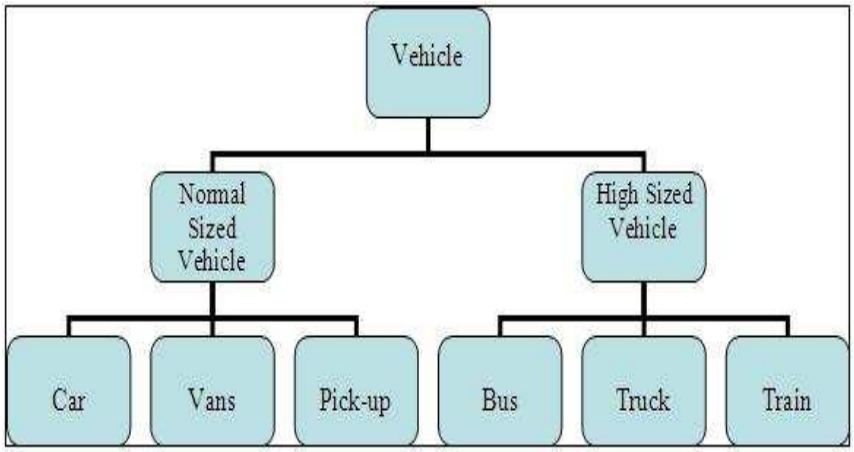


Figure 2.2: Static Taxonomy Tree for Vehicle Type

Table 2.4: A Sample Data Which Requires Prevention of Attribute Linkage

Age	**Zip Code**	**Disease**
18	06100	Viral Infection
55	06330	Cancer
60	06350	Cancer
20	06400	Viral Infection
35	06500	Heart Disease
50	06300	Cancer

2.2.4 k-Anonymity Studies

In [29] and [30], k-anonymity is presented as a formal protection model. Sweeney, provides a formal presentation of combining generalization and suppression to achieve k-anonymity in [27]. This study uses generalization hierarchies during the generalization and suppression operations. Domain generalization hierarchies are introduced for categorical attributes and value generalization hierarchies for numeric attributes. Meyerson and Williams [10] showed that k-anonymization with minimum number of suppressions is NP-hard. Aggarwal et al. [11] showed that the problem of k-anonymization is NP-hard even when the attribute values are ternary. Some approximation algorithms are proposed for this problem in [31] and [10]. Greedy heuristic algorithms are introduced in [26] and [28] in order to produce k-anonymous data while preserving the property of building decision tree classifiers. Therefore, privacy of data is guaranteed and it can be used for classification purposes.

2.2.5 Attribute Linkage vs Record Linkage

k-anonymity solutions solve the prevention of "record linkage attack" which is actually finding the owner of a record through quasi-identifier attributes. However, it is shown that without finding the exact owner of a record, if sensitive attribute exists in a record , it may be possible to identify sensitive attribute of an individual in some circumstances by an attack called "attribute linkage attack" [2].

In Table 2.4, a sample data collected by a hospital is shown. In this data, age

Table 2.5: 3-Anonymized Version of Sample Data

Age	**Zip Code**	**Disease**
18-35	06***	Viral Infection
18-35	06***	Viral Infection
18-35	06***	Heart Disease
50-60	063**	Cancer
50-60	063**	Cancer
50-60	063**	Cancer

Table 2.6: Anonymization of Sample Data with 2-Diversity

Age	**Zip Code**	**Disease**
18-50	06***	Viral Infection
18-50	06***	Viral Infection
18-50	06***	Cancer
35-60	06***	Heart Disease
35-60	06***	Cancer
35-60	06***	Cancer

and zip code are considered as quasi-identifiers and disease is chosen as a sensitive attribute. Main aim of privacy preservation is hiding disease information of an individual. Table 2.5 gives a possible 3-Anonymized version of this data. All the last three records of this table has sensitive attribute value, 'cancer'. Therefore, prevention of record linkage through k-anonymization does not solve the privacy problem. Sensitive attributes of these records are easily identified in this case. In order to address this problem l-diversity notion is introduced [7]. l-diversity requires that sensitive attribute value of a record owner is hidden among $l-1$ sensitive attributes. Table 2.6 shows anonymized data which has 2-diversity property. This property guarantees that each anonymity set has at least two different sensitive values.s

In order to prevent attribute linkage, in addition to l-diversity notion, p-sensitivity and t-closeness notions are proposed [8, 9].

2.3 Entropy Notion For Quantifying Privacy and Calculation of Information Loss

In information theory, entropy measures the uncertainty of a random variable. Quantity introduced by entropy measures the content of an information in terms of bits. Entropy notion of Shannon [32] is the theoretic formulation of entropy which is widely used in information theory literature.

Practically, Shannon entropy estimates the average number of bits for encoding strings composed of symbols with different frequency. It introduces lower bounds for compressing and storing communicated data.

Shannon's entropy is used in quantifying privacy for a privacy protection system developed for context aware services [33]. This entropy notion determines the level of abstraction in location and personal preferences reports collected by context aware service servers. Shannon entropy is used in improving l-diversity in order to prevent probabilistic inference attacks [7]. Original l-diversity method guarantees existence of l distinct sensitive attributes in each anonymity set. However, if frequency of some sensitive attributes is higher, attacker can conclude that individual has the sensitive attribute with more frequency. l-diversity notion extended by entropy gives higher value for evenly distributed sensitive attributes.

A top-down refinement algorithm is proposed in order to perform privacy preserving data publishing for cluster analysis [34]. Entropy notion is used for measurement of information loss and anonymity calculation in each refinement stage.

Entropy based anonymity measurement model is proposed for measuring the privacy level of systems protecting sender anonymity [35]. Measurement model is applied for mix based e-mail applications, Crowds and Onion Routing.

There exists clustering algorithms which uses information theoretic distance functions based on Shannon entropy [36] [37] [38].

2.4 Privacy in Wireless Sensor Networks

Anonymity is being not identifiable of a subject within a set of "subjects". In networks like Internet and Ad hoc networks, identities of communicated parties are considered as subjects. Therefore, anonymity studies concentrate on hiding sender and/or receiver identity information. However, in sensor networks subject is mostly event information. Event information is sensed and sent to sinks or other central storage node. Anonymity problem of event information can be dealt in sensing and sending stages. In the literature, for sensing stage, data aggregation is used for creating anonymous event data. For communication stage, variations of known anonymity techniques are applied.

In Gruteser and Grunwald's study [39], an anonymity solution is proposed for providing high degree of privacy in location based services. They assert that adversary can get the location information of an individual by different types of attacks like eavesdropping of whole traffic or compromising the server provider's system. They claim that adversary can identify a specific individual by linking the event location information with a priori knowledge about the event. If location information is disseminated continuously for tracking purpose in applications like traffic monitoring, fleet management and 'pay as you drive' insurance, adversary can track all movements of an individual. Location and time information of events are cloaked so that an outsider cannot differentiate any individual among the other k different individuals. Event messages are sent to anonymity server in which any identifiers like network addresses are removed, data perturbation is performed and reordering of the incoming messages from different nodes is accomplished. This study omits the threat that adversary who captures the traffic between nodes and anonymity server can do deeper traffic analysis.

Data cloaking and a communication anonymity solution is used together to have an anonymous system in [40]. Proposed sensor network acts as an in-building occupant movement tracking system in which the main purpose is finding the popularity level and usage amount of different parts of a large building. The main aim is pro-

viding privacy through anonymity. Sensor network has a hierarchical structure and do data cloaking operations on sensed data or sensor node ID information. Data cloaking is done at the nodes of upper hierarchies. With less spatial accuracy or data perturbation, they tried to have an anonymous system. In order to prevent eavesdropping and other active attacks, node communications are encrypted and authenticated by SPINS like protocols [41]. Prevention of traffic analysis is tried to be accomplished by periodic message sending independent from sensor readings. This study presents a solution for data perturbation tolerant applications. However data perturbation may not be tolerable in many critical applications. On the other side, adversary can understand whether cloaking of sensor node ID information is done and he can find the actual sensor node ID by following the traffic in hop by hop basis. Therefore, it is not a good solution in environments where global eavesdropping threat exists.

Data aggregation is also studied by Przydatek, Song and Perrig [42]. In this study some nodes are chosen as aggregators and they collect data from sensor nodes. Aggregator node does aggregation operation and proves that the aggregation result is consistent by random sampling and interactive proof mechanisms. Aggregation is done for securely finding minimum, maximum and average sensing values.

Ozturk et al. [43] proposed *phantom routing* method for hiding location information of originator sensor node. This routing method is alternative for flooding type routing algorithms. Threat model bases on the existence of only one adversary node in the sensor environment. Sensor network tracks moving objects and send appropriate event information to the sink. Adversary eavesdrops the traffic on this node and can determine the previous hop of routing messages. Adversary tries to reach a moving target object. After detecting the previous hop of routing message, he goes to near of that hop and get closer to the target. He continues this operation until the target is caught. Proposed routing algorithm aims to make this catching operation difficult by hiding the location of sensor node. Although the study presents a good solution to the problem, threat model consists of weak assumptions that there exists only one adversary node in the system. It does not assume the

existence of global eavesdropper threat.

Wadaa et al. [44] studied on providing anonymity of coordinate system, cluster and routing structures during the network setup phase of a wireless sensor network. The study is lack of formal methods those prove the provided anonymity level and there is no experimental results showing the effectiveness of the proposed system. Also the proposed solution is restricted to the network setup phase.

Castelluccia et al. [45] proposed *homomorphic encryption* method that securely aggregates sensor findings in an energy efficient way. This work deals with the aggregation functions, which compute average or variance of sensor findings. Protection of location privacy is guaranteed by k-anonymity in location based services those are given on mobile networks [46]. In this work, each mobile client specifies a minimum level of anonymity and maximum temporal and spatial tolerances. Proposed methods try to provide the needed anonymity level within these quality of service parameters.

2.5 Privacy in Sharing of Security Logs

Some organizations implement intrusion log collection systems for determination of general security level of the Internet. They aim to provide early warning systems about security threats. Deepsight Threat Management System [47], which is managed by Symantec, gives information to its customers about the emerging threats, vulnerabilities, risks, workarounds and other references. This system collects logs from intrusion detection systems, virus scanners and firewalls. System does not use any anonymization method during data collection. Data is not shared with the research community; it is used for commercial purposes. Internet storm center, which is implemented by SANS [48], is volunteered and non-commercial version of DeepSight. It collects intrusion detection system and firewall logs from volunteer organizations and produces general analysis results for public and creates customized warning information for organizations. Internet storm center uses Dshield distributed detection system for data collection and analysis. The imple-

menters state that they remove the identifying parts of intrusion data by masking destination IP of logs. However, this anonymization cannot guarantee to prevent the attack types those are similar to re-identification attacks proposed in privacy preserving data publishing area [6].

There exist studies about anonymizing IP address of network logs. Actual IP addresses are replaced by a randomly selected IP addresses according to a permutation function. New random IP addresses do not contain even the sub-net information. Entity who captures randomized IP addresses can only deduce that the logs having the same IP addresses are actually originated from the same host.

Truncation is another anonymization method that converts fixed number of least significant bits of IP address to zero. This means, the remaining information can show only subnet or network class information of IP addresses. From anonymized data, anyone can deduce the subnet information but cannot determine whether logs belonging to a particular subnet are originated from the same host or from many hosts.

In *prefix-preserving pseudonymization*, which is adapted in *TCPdriv* [49], IP addresses are mapped to pseudorandom anonymized IP addresses by an anonymization function that uses common tables. If first k-bits of original IP addresses are common, function produces anonymized outputs which are also common in terms of first k-bits. Xu. et al. [50] proposed a prefix-preserving pseudonymization method, Crypto-PAn that works consistently in multiple traces by using a shared key. This study also includes the security evaluation of prefix-preserving pseudonymization schema. Slagell et al. [51] re-implemented Crypto-PAn in Java for anonymization of Netflow logs. Netflow logs are logs of network traffic generated by routers. These logs have a standard log format. Slagell et al. also embedded their own key generator to their implementation.

Zhang et al. [52] studied on the anonymization of all fields of Netflow and syslog data for sharing them with managed security service providers. Syslog is s standard log format used by Unix or Linux based systems. This study gives brief information about the known anonymization techniques, which can be used for anonymization of

IP addresses, time information and port numbers. Time information is anonymized by random time shift method . In this shifting method, the time of log is changed with random number, but the relative time intervals between related logs are preserved. Common public port numbers remain in the logs, however sensitive ports of sensitive hosts are anonymized.

Studies about privacy preserved security logging generally focus on changing the truthfulness of log attributes by pseudonymization or by other anonymization techniques at record level. These techniques may help to do some basic searching activities or may help to perform some limited types of analysis on privacy preserved logs. However, the idea of privacy preserving data publishing has not been adapted to this field. Intrusion log collection systems like DeepSight or Internet Storm Center need logs with higher data utility. Chapter 5 of this thesis adapts privacy preserving data publishing techniques to intrusion log collection systems. Proposed method collects data, which have higher data utility, while preserving privacy of log owners.

Chapter 3

Proposed k-Anonymization Clustering Method(k-ACM)

Solving the k-anonymity problem is proved as an NP-Hard [10,11], therefore various heuristic methods have been developed to minimize data loss as much as possible [6, 28]. In this thesis, we proposed a framework based on the k-anonymization method, k-ACM, that solves the k-anonymity problem by a bottom-up hierarchical clustering algorithm. The basic clustering notion is derived from UPGMA (Unweighted Pair Group Method with Arithmetic Mean) [53].

UPGMA is based on the idea of iteratively joining two closest clusters until one cluster is left. A suitable distance definition has to be done to measure the distance between any two clusters. All the 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 considered as an individual cluster. Closest two clusters are found and combined into one common cluster. Distances of newly formed cluster to the other clusters are recalculated and distance matrix is updated. The same work continues until one cluster is formed.

k-ACM is applied to the data portion containing only the quasi-identifier attributes. The basic idea is to partition the data vectors into clusters where each cluster has at least k vectors. After the clustering, vectors in one cluster are anonymized to a common vector, named representative vector which is actually the k-anonymization output of all vectors in that cluster. All quasi-identifier attributes of the input data is replaced by the corresponding attributes of the representative vector. The clustering process ensures that similar vectors are grouped in clusters

so that their anonymization does not cause a significant data loss.

Running time of k-ACM is found as $O(n^2 log(n))$ where n is the number of event records. The details about the derivation of running time is given in Section 3.8.

In the later chapters of thesis, it is shown that k-ACM presents an efficient baseline in anonymizing data for multiple receivers having different privacy levels.

Distance function of clustering process has to be formed in order to reach to efficient adaptation of clustering idea to the problem. In k-ACM, cluster decisions are made according to the amount of information loss occurred during cluster combinations. Distance function used in k-ACM calculates information loss by using entropy notion.

In order to choose the appropriate value of parameter k, information loss and the anonymity level of data have to be quantified. By appropriate quantification methods, trade-off analysis between data utility and privacy level can be made efficiently. In this thesis, entropy based quantification methods are proposed for these two metrics.

In this chapter, first four sections cover the baselines of k-ACM. Section 3.1 explains the proposed dynamic taxonomy tree idea for generalization methods. Section 3.2 introduces the information loss metric that is used in evaluating k-ACM results. Section 3.3 presents the distance function used in the proposed method. Section 3.4 proposes entropy based anonymity measurement method. Section 3.5 gives the proposed algorithm k-ACM. Complexity analysis of k-ACM is given in Section 3.8. The notation used in this chapter is given in Table 3.1.

3.1 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

Table 3.1: Notation Table For k-ACM

Notation Explanation	**Notation**
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)	$I^h_u (I^h_u = D^h[s][t])$

Table 3.2: 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

of using static taxonomy tree in generalization of categorical attributes. For example, in the static taxonomy tree given in Figure 2.1, generalization of 'Buket Street' and 'Selvi street' values yields to 'Istasyon Avenue' value. However, this causes information loss since records having Istasyon Avenue may also include 'Serin Street'. In order to solve this over-generalization problem as much as possible, we propose to use *dynamic taxonomy tree* instead of static one. In the proposed dynamic taxonomy tree model, the tree is dynamically updated by creating new internal nodes (i.e. attribute values) during generalization and in 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 3.1, in our dynamic approach generalization of 'Buket Street' and 'Selvi Street' causes a new categorical value with name 'Buket Street- Selvi Street' to be generated instead of generalizing to existing 'Istasyon Avenue'. This new value means the attribute is either 'Buket Street' or 'Selvi Street', but not 'Serin Street'.

For example, if we applied dynamic taxonomy tree method to the sample data given in Table 2.2, 2-anonymized output in Table 3.2 is obtained. From this anonymized data, the number of vehicles in each street and the number of total vehicles in each type can be calculated accurately. These calculations cannot be accurately done from the anonymized data in Table 2.3.

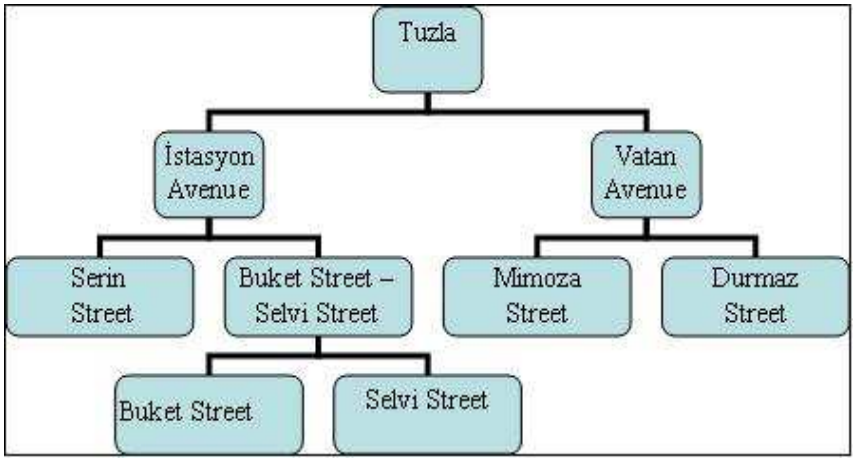


Figure 3.1: A Sample Node Addition in Dynamic Taxonomy Tree

In order 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. In order 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) has 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 value '1' increases as generalizations occur. Bit string having many bits having value '1' actually represents an internal node.

For a numerical attribute, range of the attribute can be divided into intervals where each interval has suitable equal range size. The size of numerical attribute's bit string is set to number of intervals. Each interval corresponds to a distinct bit and if an attributes belongs to an interval, corresponding bit of the interval is set to one in the bit string. The number of intervals can be determined according to

the accuracy need for that attribute. More accuracy need means more number of intervals. Increase in the number of intervals enlarges the sizes of messages, so the needed transmission energy. Therefore a balance between energy and accuracy must be constructed in choosing the number of intervals.

Suppose input data is a table T having m attributes, 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 bit string representation of j'th attribute of i'th record. k'th bit of B_{ij} is shown as $B_{ij}(k)$. Suppose that j'th attribute of table is categorical and there are d_j distinct values. These values are indexed by k and shown as $V_j(k)$where $\{k : 1 \leq k \leq d_j\}$. Bit string of this categorical attribute has a size of d_j and 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 attribute is numerical, the range of attribute is divided into equal-sized intervals. Assume that j'th attribute is numeric and attribute range is divided into e_j number of intervals. Each interval is indexed by k. Bit string representation of this numeric attribute has a size of e_j and 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, anonymizing entries 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 tree. During the formation of dynamic taxonomy tree, 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.

Table 3.3: A Sample Bit String Representation Set

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

Table 3.4: A Sample Normalized Version of Bit String Representation Set

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

3.2 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 [37]. The difference of entropies 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 3.1 and C is the random variable that gets the probability value 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 is normalized according to the number of bits having value '1' in that entry (from now on we refer "true bit" to a bit having value '1') and normalized version forms data set $\overline{B}$. A sample data set is shown in Table 3.3. Here, there are two records; each record has three attributes; each attribute is categorical and each has five distinct attribute values. Table 3.4 shows the normalized version of 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, conditional entropy gives the uncertainty about the prediction of the original attribute values of a record when we have the knowledge of corresponding

k-anonymous bit strings of that record. 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 each attribute value is represented by an additional bit. Therefore, if an entry has only one true bit, that entry does not have information loss. In this situation, we have no doubt that this true bit is the true bit that comes from the original data. As the number of true bits increases, disorder of the data increases because it is harder to predict which one of them is the original true bit. Prediction gets harder because information is lost due to the increase in the number of true bits. Conditional entropy, which is used in order to calculate the disorder of the data, is a well measurement tool for the information loss. Conditional entropy $H(C \mid B)$, which is equal to information loss of table T, $IL(T)$, can be found as follows:

$$IL(T) = H(C \mid B) = \sum_{B_{ij} \in B} p(B_{ij}) H(C \mid B = B_{ij}). \tag{3.1}$$

$$IL(T) = - \sum_{B_{ij} \in B} p(B_{ij}) \sum_{k \in \{1..z\}} p(C = k \mid B_{ij}) \log p(C = k \mid B_{ij}). \tag{3.2}$$

In Equation 3.2, it is assumed that each attribute is converted to bit strings having size z. This means 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. C random variable can take values from the set $\{1..z\}$. Actually, $\overline{B}$ is calculated for finding the value of this random variable.

$$p(C = k \mid B = B_{ij}) = \overline{B}_{ij}(k) \; for \; each \; k : \; 1 \leq k \leq z. \tag{3.3}$$

In Equation 3.2, it is assumed that each record has equal probability to be chosen and

each attribute of record has the same probability, therefore probability mass function of j'th attribute of i'th record, $p(B_{ij})$, is calculated as $p(B_{ij}) = \frac{1}{m.n}$. Equation 3.2 can be rewritten as follows:

$$IL(T) = H(C \mid B) = -\sum_{B_{ij} \in B} \frac{1}{m.n} \sum_{k \in 1..z} \overline{B}_{ij}(k). \log \overline{B}_{ij}(k). \tag{3.4}$$

Suppose that F is the array that contains the number of true bits of the bit string array B. Total number of true bits in B_{ij} is F_{ij}. Total number of elements in $\overline{B}_{ij}(k)$ that has the value of $\frac{1}{F_{ij}}$ is equal to F_{ij}, and the rest is zero. Therefore, the second sum operation of Equation 3.4 yields the value, $\log \frac{1}{F_{ij}}$. The simplest equation for the information loss of data table T, $IL(T)$, can be calculated as follows :

$$IL(T) = H(C \mid B) = -\sum_{F_{ij} \in F} \frac{1}{m.n} \log \frac{1}{F_{ij}} = \frac{1}{m.n} \sum_{F_{ij} \in F} \log F_{ij}. \tag{3.5}$$

3.3 Distance Calculation

The aim of our method is minimizing the 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 has to choose the most suitable cluster pair, which creates minimum information loss when they are combined. To do so, a suitable distance calculation method is needed. In Section 3.2, conditional entropy notion is used in calculating the overall information loss of k-anonymized data. This notion is adapted in calculating the distance between any two clusters such that distance between any two clusters is the entropy loss caused by merging them.

At the h^{th} iteration the distance between s^{th} and t^th clusters is defined as $D^h[s][t]$. Suppose that resulting cluster after merging of clusters s and t is represented as cluster u in iteration $h+1$. Cluster u has $|V_s^h + V_t^h|$ number of elements. Merging

operation means that $|V_s^h|$ number of input vectors having value R_s^h and $|V_t^h|$ number of input vectors having value R_t^h is converted to $|V_s^h + V_t^h|$ number of vectors having value R_u^{h+1}. Conditional entropy value before merging operation is represented as E_{st}^h and computed by the help of Equation 3.5 as follows:

$$E_{st}^h = \frac{1}{m.(|V_s^h| + |V_t^h|)} |V_s^h| \sum_{i \in \{1..m\}} \log(F(R_s^h[i])) + |V_t^h| \sum_{i \in \{1..m\}} \log(F(R_t^h[i])). \quad (3.6)$$

E_u^{h+1} is the conditional entropy value after merging operation and calculated as follows:

$$E_u^{h+1} = \frac{1}{m.(|V_s^h| + |V_t^h|)} (|V_s^h| + |V_t^h|) \sum_{i \in \{1..m\}} \log(F(R_u^{h+1}[i])). \quad (3.7)$$

$$E_u^{h+1} = \frac{1}{m} \sum_{i \in \{1..m\}} \log(F(R_u^{h+1}[i])). \quad (3.8)$$

Distance between cluster s and t at iteration h, $D^h[s][t]$, is calculated as follows:

$$D^h[s][t] = E_u^{h+1} - E_{st}^h. \quad (3.9)$$

3.4 Anonymity Measurement

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

Suppose that A and B are the sets of records in original data and k-anonymous data respectively. Conditional entropy, $H(A \mid B)$, is used as anonymity measurement method. $H(A \mid B)$ gives the uncertainty level of prediction of the record in A when the corresponding anonymous version of the record in B is known. Here, more

uncertainty means high anonymity level. The amount of anonymity, Q, is calculated as follows:

$$Q = H(A \mid B) = \sum_{b \in B} p(b).H(A \mid B = b) = -\sum_{b \in B} p(b). \sum_{a \in A} p(a \mid b). \log p(a \mid b). \tag{3.10}$$

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 data is exactly k-Anonymous. In other words, suppose that anonymous data has n records and each record has exactly the same quasi-identifier record set as some other $k-1$ 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 $n-k$ ones. By evaluating Equation 4.13 with these values, we calculate the minimum anonymity level of a k-Anonymous data, which is denoted as Q_{min}, as follows:

$$Q_{min} = log(k). \tag{3.11}$$

On the other hand, the purpose of k-ACM is not to form clusters having exactly k elements; it is 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 of 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:

$$Q = -\sum_{C_F^i \in C_F} \frac{1}{s(C_F^i)}. \log \frac{1}{s(C_F^i)}. \tag{3.12}$$

Uncertainty gets the lowest value when each cluster has exactly k elements. However, uncertainty increases when clusters have different number of elements and number

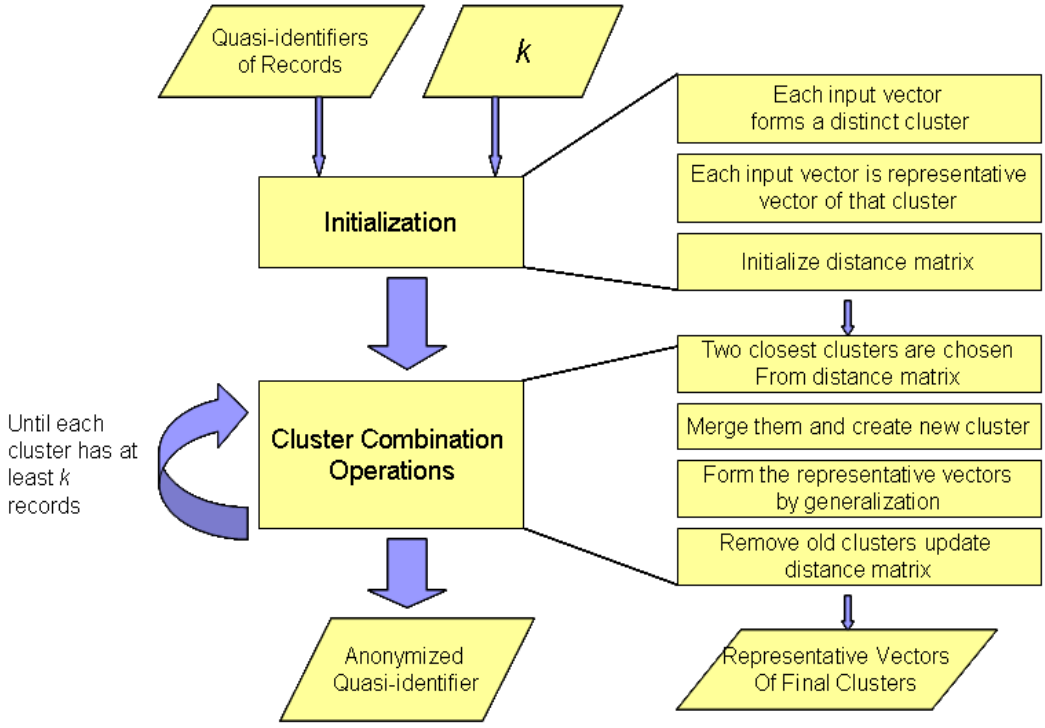


Figure 3.2: General Flowchart of k-ACM

of clusters gets lower. Therefore, the inequality, $Q \geq Q_{min}$, holds for every possible output of k-ACM.

3.5 k-Anonymous Clustering Method (k-ACM)

Our method, k-ACM, starts with the initialization phase where a new cluster is created for each input vector. After the initialization, clustering operation starts and clustering is performed. The general flowchart of the method is shown in Figure 3.2 and algorithm of k-ACM is given in Algorithm 3.2.

In the k-anonymization stage, k-ACM forms the clusters in a bottom-up fashion

Algorithm 3.1 Function Cluster Combination

Input: parameter, k, distance matrix, D^h
Output: New cluster, L_u^h, updated distance matrix, D^{h+1}
1: Find clusters, L_s^h, L_t^h, having minimum distance in distance matrix D^h
2: create a new cluster L_u^{h+1}
3: $V_u^{h+1} \leftarrow V_s^h \bigcup V_t^h$
4: $|V_u^{h+1}| = |V_s^h| + |V_t^h|$
5: **for** each i^{th} bit string of representative vector **do**
6: $\quad R_u^{h+1}[i] \leftarrow R_s^h[i]$ **or** $R_t^h[i]$
7: **end for**
8: Remove clusters, L_s^h, L_t^h
9: Find the distance of L_u^{h+1} to other clusters, update D^{h+1}

Algorithm 3.2 Main Function of **k**-ACM

Input: Table, T, number of records, n, number of attributes, m, anonymization parameter k
Output: k-anonymized table, k-ACM(T)
{Initialization}
1: $h = 1$
2: **for** all i where $\{i : 0 < i < n\}$ **do**
3: $\quad$ Create cluster array, $\{L_i^1\}$
4: $\quad$ Add record, T_i to V_i^1
5: $\quad$ Set initial size of cluster, $|V_i^1| = 1$
6: $\quad$ Initialize the representative vector, $R_i^1 \leftarrow T_i$
7: $\quad$ Initialize the distance matrix D^1 by using Equation 3.9
8: **end for**
{k-anonymization}
9: **while** not for each cluster $|V_i^h| \geq k$ **do**
10: $\quad$ Call Function ClusterCombination (k, D^h) given in Algorithm 3.1.
11: $\quad$ h=h+1
12: **end while**
{Form the output of k-ACM}
13: k-ACM(T)$\leftarrow \varnothing$
14: **for** each c̄luster, L_s^h in L^h where $\{s : 0 < s < |L^h|\}$ **do**
15: $\quad$ Append R_s^h and $|V_s^h|$ to k-ACM(T)
16: **end for**

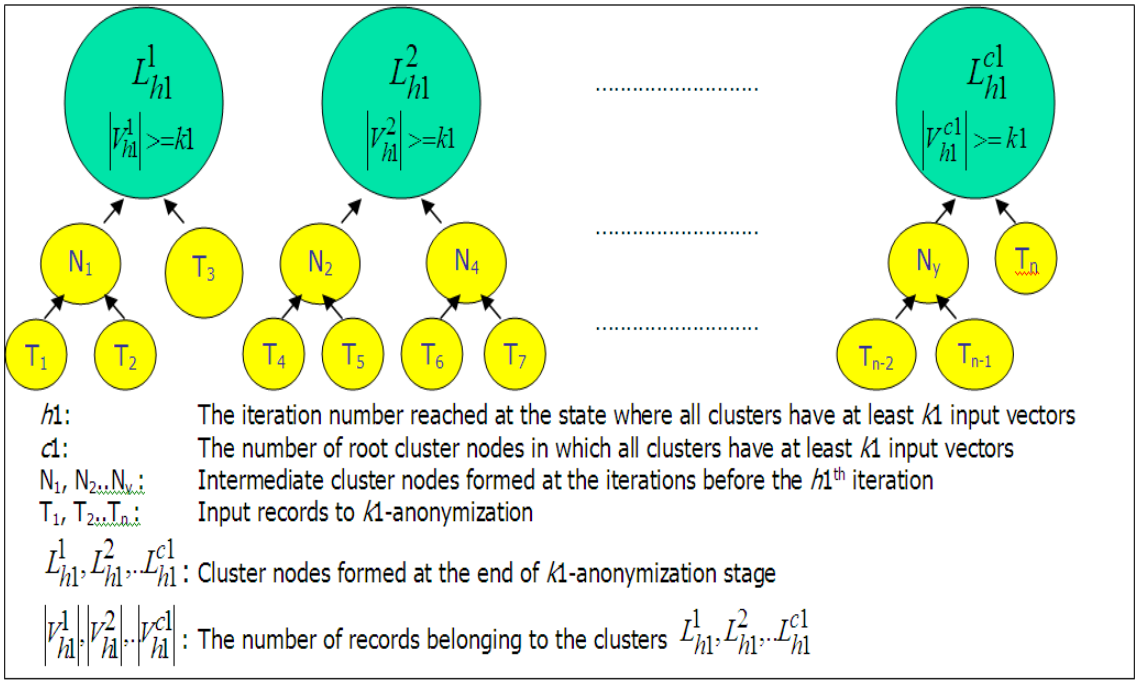


Figure 3.3: A sample tree structure of clusters obtained at the end of k-anonymization

like UPGMA until each cluster represented has at least k records. A sample tree structure of clusters obtained at the end of the k-anonymization stage is shown in Figure 3.3. Here, $h1$ is defined as the iteration number needed to complete the k-anonymization stage. In this tree structure, each tree node represents a cluster and cluster size is the number of records that belong to the corresponding cluster. This tree has $c1$ root nodes, identified as $L^{h1}_1..L^{h1}_{c1}$, and their sizes are at least k.

In each cluster combination operation, the closest clusters are found and a new cluster, which contains all the vectors belonging to the chosen closest clusters, is formed. Distance calculations are done according to Equation 3.9. Representative vector of the new cluster is bitwise OR of representative vectors of child nodes. Here, OR operation acts as a generalization operation and the clusters in higher tree levels have more generalized representative vectors. A sample case for cluster combination operation is shown in Figure 3.4. Suppose that the closest clusters

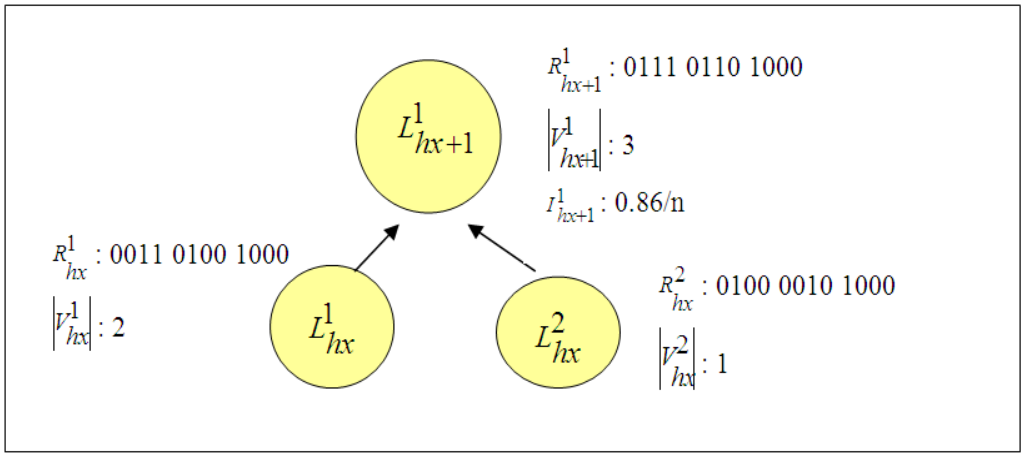


Figure 3.4: A sample case for forming new cluster by combination of two closest cluster

in the hx iteration are L_1^{hx}, L_2^{hx} and their representative vectors 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, all of them are categorical and each categorical attribute 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. Sizes of clusters, L_1^{hx}, L_2^{hx} are represented as $|V_1^{hx}|$ and $|V_2^{hx}|$ respectively. The size of new cluster, $|V_1^{hx+1}|$, is the summation of the sizes of 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}.

3.6 Termination Proof of k-ACM

Assume that at iteration h, the set of clusters is labelled as L^h, the total number of clusters is $|L^h|$ and i^{th} cluster is represented as L_i^h. The number of records is n. k is the anonymity variable where $k < n$. Representative vector of cluster, L_i^h, is R_i^h. The number of elements is denoted as $|V_i^h|$.

Lemma 1: If clusters, L_i^{h1} for all $i \in \{1..|L^{h1}|\}$ at iteration $h1$ of k-ACM, have at least k items, $h1$ is the last iteration of k-ACM and the output of k-ACM has k-anonymity property.

Proof:

k-ACM sends the representative vectors, R_i^{h1} for all $i \in \{1..|L^{h1}|\}$ with $|V_i^{h1}|$. Since k-ACM guarantees the holding of inequality $|V_i^{h1}| \geq k$ for all $i \in \{1..|L^{h1}|\}$, each cluster forms an anonymity set having size k. *Q.E.D.*

Lemma 2: Maximum number of clusters at the end of k-ACM is (n/k). This situation can be reached when all clusters have exactly k records (For simplicity, assume that n is divisible by k).

Proof:

Assume that at iteration $h1$, k-ACM is terminated. Also assume that the Equation 3.13 is hold.

$$|L^{h1}| > (n/k). \tag{3.13}$$

From Lemma 1, it is known that each cluster has at least k items. For total number of records in all clusters, n, it can be stated that $n \geq k.|L^{h1}|$. By using Equation 3.13 we can conclude that $n \geq k.|L^{h1}| > n$ but a contradiction is reached. We conclude that assumption about Equation 3.13 is wrong.

For the second part of Lemma 2, assume that for i^{th} cluster, $|V_i^{h1}| > k$. Since all other clusters has at least k records, the following inequality is hold: $n - |V_i^{h1}| \geq ((n/k)-1)k$. By adding each side of both inequalities we conclude that $n > n$ which is contradiction. Therefore, assumption is wrong. *Q.E.D.*

Theorem 1: k-ACM terminates after at most $n-1$ iterations.

Proof:

In each iteration of k-ACM, two closest clusters are chosen and merged. It is guaranteed that in each iteration h, the number of clusters is equal to $|L^h| =$

$|L^{h-1}| - 1$

Initially, for clusters L_i^1 where for all $i \in \{1..|L^1|\}$, $|V_i^h| = 1$ and $|L^1| = n$. Bottom-up clustering can continue until one cluster left. Suppose that at iteration s, there is only one cluster L_1^s. For the size of cluster, following equality holds: $|V_1^s| = n$. Since $n > k$, and by using Lemma 1, we can conclude that the output of k-ACM has k-anonymity property. Since in each iteration, the number of clusters is decreased by one, total number of iterations is $n - 1$. *Q.E.D.*

Theorem 2: k-ACM terminates after at least $n-(n/k)$ iterations (for simplicity, it is assumed that n is divisible by k).

Proof:

Initially number of clusters, $|L^1|$ is equal to n. In each iteration number of clusters is decreased by one. According Lemma 2, the number of clusters produced by k-ACM is maximum when each cluster has exactly k items. Therefore, maximum number of clusters is equal to n/k. We can conclude that the minimum number of iterations for decreasing n clusters to n/k clusters is $n - (n/k)$. *Q.E.D.*

3.7 Worst Case Information Loss Analysis of k-ACM

Assume that k is the anonymization parameter and also assume that data has n records and m attributes where each attribute i has p_i different attribute values. T is the input data set for anonymization. $IL(T)$ is the information loss value of data set T.

Lemma 3: k-ACM generates clusters having at most $2k - 1$ records.

Proof:

In k-ACM, if any s^{th} cluster of $(h)^{th}$ iteration, L_s^h, has a size $|V_s^h| > k$ then cluster L_s^h does not involve in any combination operation in later steps. If a cluster, L_u^{h+1},

is a result of combinations of two clusters, L_s^h and L_t^h, we can say that $|V_s^h| \leq k$ and $|V_t^h| \leq k$. If both clusters have exactly k records they are not combined. Therefore, this requirement can be revised so that $|V_s^h| \leq k$ and $|V_t^h| < k$. Since, $|V_u^{h+1}| = |V_s^h| + |V_t^h|$ then we can conclude that $|V_u^{h+1}| < 2.k$. Maximum value for the number of records belonging to one cluster is $2k - 1$. *Q.E.D.*

Theorem 3: If for each p_i where $i \in \{1..m\}$, $p_i = p$ and $p < k$, information loss is $log(p)$ in worst case.

Proof:

According to Equation 3.5, information is average of log of true bits in each data entry. Representative vector of s^{th} cluster in h^{th} iteration is represented as R_s^h and i^{th} entry is Representative vector of s^{th} cluster in h^{th} iteration R_s^h. $R_s^h[i]$ represents widest range when at least one record exists with each possible attribute value. Since in output of k-ACM a cluster has at least k records, where $p < k$, it is possible to have p distinct attributes in each cluster. In this situation, length of i^{th} bit string, $R_s^h[i]$, for each i where where $i \in \{1..m\}$, is p and each bit is actually true bit. Therefore, information loss is $\log(p)$ in worst case. *Q.E.D.*

Theorem 4: If for each p_i where $i \in \{1..m\}$, $p_i = p$, $h1$ is the final iteration and $p \geq 2k - 1$, information loss is $log(2k - 1)$ in worst case.

Proof:

From lemma 3, a cluster, L_s^h has size $|V_s^h|$ where $|V_s^h| \leq 2.k-1$. Since $p \geq 2k-1$, widest value range of i^{th} bit string of R_s^{h1}, for each i where where $i \in \{1..m\}$, can be set when each record in the cluster has different attribute value. In this situation, each i^{th} bit string of representative $R_s^{h1}[i]$ has a length of p and at most $2k - 1$ of them are true bits. According to Equation 3.5, information loss is $log(2k - 1)$ in worst case when $p \geq 2k - 1$. *Q.E.D.*

Theorem 5: If for each p_i where $i \in \{1..m\}$, $p_i = p$, $h1$ is the final iteration and $k \leq p \leq 2k - 1$, then information loss is found as follows:

$$IL(T) \leq 1/n. \sum_{s \in L^{h1}} |V_s^{h1}|.log(\min(|V_s^{h1}|, p)) \tag{3.14}$$

Proof:

In each cluster, L_s^{h1}, any i^{th} bit string $R_s^{h1}[i]$ has a length of p. Maximum information loss occurs when each record has different attribute value. Since $|V_s^{h1}| < p$, $|V_s^{h1}|$ of p bits can be true bits at most. If $|V_s^{h1}| > p$, maximum information loss occurs when all attribute values are covered by the records of cluster. All bits of p bits are true bits in this situation. If we combine these two conditions, we can conclude that number of true bits are determined by $\min(|V_s^{h1}|, p)$ in each bit string of representative vectors. Therefore maximum information loss is calculated as $log(\min(|V_s^{h1}|, p))$. Total maximum amount of information loss of a cluster, V_s^{h1}, is $m.|V_s^{h1}|.log(\min(|V_s^{h1}|, p))$. Calculation of maximum information loss belonging to all clusters is $\sum_{s \in L^{h1}} m.|V_s^{h1}|.log(\min(|V_s^{h1}|, p))$. information loss of table T $IL(T)$, is calculated by the average information loss per data entry according to Equation 3.5. Therefore, upper bound for $IL(T)$ is $IL(T) \leq 1/n. \sum_{s \in L^{h1}} |V_s^{h1}|.log(\min(|V_s^{h1}|, p))$. *Q.E.D.*

3.8 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-identifier and each attribute has distinct V different attribute value. Initialization phase mainly calculates the initial distance matrix and the running time of this part is $O(n^2.m.V)$. Initially there are n clusters and at the end of k-anonymization phase, the minimum number of clusters is n/k. Therefore, $n - n/k$ cluster combination operation occurs. Cluster combination consists of finding the minimum distance in the distance matrix and matrix reorganizing so that the distance values of new cluster are added and distance

values of previous clusters are removed. If binary heap structure is used for finding minimum distance, formation of initial min 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 heap and adding the distances of new cluster to the heap need $2n$ deletion and n addition operations which cost $O(n \log(n))$. Reorganization of distance matrix can be done in $O(n.m.V)$ time sequentially with maintaining the heap. As a result, cost of each cluster combination operation is $O(nlogn + nmV)$. Recall that maximum number of cluster combination operations is $n - n/k$, the algorithm reaches to the end of k-anonymization phase in $O(n^2logn + n^2mV)$. Output enlargement for partial encryption and formation of k-ACM output takes $O(n)$ time. Totally, k-ACM takes $O(n^2logn + n^2.2mV)$. m and V generally have lower values so they can be assumed as a constant factor. The running time can be fine-tuned to $O(n^2logn)$.

Chapter 4

k-Anonymity based Framework for Privacy Preserving Data Collection in Wireless Sensor Networks

Results of advances in sensor and wireless technology, wireless sensor networks (WSNs) emerged as an important information gathering system from wide areas. They are widely used for observing many physical phenomenons of world like temperature, humidity etc. As wireless sensor technology takes progress, missions of WSNs get complicated so that they are used in human, enemy, habitat, structure or traffic monitoring applications. With the advent of wireless body are networks, applications for health monitoring of patients outside the hospitals or home-caring of elderly people have designed and implemented widely.

In recent sensor network applications, especially in object monitoring applications, the collected information is not an aggregated value like average temperature or humidity value of a region; it may be about specific individuals or specific events so that the privacy of each event information gets important. Moreover, the data gathered in these and other sensor network applications may contain several attributes for an entity. For example, traffic monitoring applications collect velocity, direction and size information of a vehicle in addition to spatio-temporal information. Collection of these attributes enables to launch re-identification attacks even in the case where the identities are withheld.

In some sensor network applications, sensor nodes may deployed over grounds

which are not physically protected; they exchange data without a fixed routing structure; they try to gather various type of information from uncontrolled areas and transfer them to the sinks in controlled or semi-controlled areas. Eavesdropping and compromising of sink are major threats in such an un-controlled environment. In addition to these threats, sink itself may not be considered as a fully trusted party.

As the complexity of wireless sensor applications increase, structure of WSNs have evolved in order to meet the new application requirements. WSNs generally have many-to-one structure so that sensors collect event information from the area and send to a unique sink. Some recent sensor applications have begun to use many-to-many structure, which actually means there exist multiple sinks in the deployed environment. WSN applications may need to send the same event information to different sinks rather than a unique sink. For example, in a home-caring application for elderly people, information about the elderly person can be sent to a family member and to a nurse at the same time.

As information collection capability of WSNs are enhanced, privacy preserving is getting one of the major problems in these networks. Huge amount of information about an individual is collected and distributed. On the other side, individuals generally need to restrict the details of personal information for privacy preservation. Therefore, countermeasures for privacy threats have to cover the both needs, enabling data collection and restricting the storage of some private parts. Countermeasures have to be designed so that they present protection for the threats of WSN environment and take many-to-many WSN structures into consideration.

On the other side, in most of the WSNs, minimization of energy consumption is one of the primary criteria due to limited battery capacity or unavailability of battery replacements. All other security countermeasures as well as the privacy preserving solutions have to perform their works with minimum energy.

In this chapter, proposed privacy framework is adapted to WSN applications for two different threat models. First threat model consists of un-trusted eavesdropper and semi-trusted sink. Here, semi-trusted notion is used for stating the level of privacy requirement. Semi-trusted sink means users are reluctant or required to share their data with sink but they do not totally trust to this data collector point. Level of privacy requirement of sink is lower than requirement of un-trusted eavesdropper.

The second threat model states that there exists multiple sinks each having different privacy levels. k-ACM is modified for each threat model. The modifications and performance analyses of each solutions are presented in Sections 4.1 and 4.2, respectively. In both sections, k-ACM is modified so that the same data guarantees the different privacy requirements of all threats. Proposed privacy solutions aim to minimize energy consumption of WSN. They successfully create a trade-off mechanism between information loss and energy consumption.

4.1 Semi-trusted Sink and Un-trusted Eavesdropper

One of the major threats in wireless medium is eavesdropper threat. During the transmission of gathered data to sink or central server, adversary having eavesdropping capability can sniff the network and get the event information. Since sensed area is uncontrolled, adversary can use his own systems to collect extra information from the area, join his knowledge with the sniffed event information and determine the attributes of specific events, such as location and time. In WSN applications of enemy tracing, habitat monitoring, traffic monitoring and human tracking [54], eavesdropping threat have to be dealt with so that the privacy of data during the transmission has to be provided.

The privacy problem is not limited to the threat of eavesdropper. In some

situations, the data shared with 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 sink effects the whole system since it stores all event information.

In some other applications, sink itself may not be considered as a fully trustable entity by the WSN users. 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 central server to know their exact spatio-temporal information in non-urgent times. Therefore, it is needed to provide privacy of personal information shared with the sink. People using WSN applications like caring systems for elderly people or smart home monitoring systems may need protection of their privacy from the parties where they share their personal information.

Trusted data collection model used in privacy preserved data publishing methods [2] does not fit directly to WSN environment since the data shared entity, sink, may not be fully trusted and there may be other un-trusted parties like eavesdroppers. Trusted entities where anonymization would be done have to be distributed so that compromising of a trusted entity does not lead to loosing all data of system. A new data collection model for a WSN environment has to be adopted.

The threat model is based on the threat due to untrusted eavesdropper and threat due to semi-trusted sink. Therefore, 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 $k2 \geq k1$.

Designers of privacy preserved data collection system for WSN have to concentrate on reducing the energy cost. Studies show that [55], energy consumption is heavily dependent on transmission/reception of data packets. Therefore, shortening

the size of event information plays a crucial role in energy reduction. In proposed method, shortening of message sizes are accomplished by sending the common quasi-identifier attributes of events only once to sink instead of sending the same data iteratively for each event.

Encryption is introduced as an anonymity operation in addition to generalization. Encryption enables to provide different privacy levels for eavesdropper and sink. Appropriate encryption keys are shared between sink and sensor nodes where anonymization takes place.

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

4.1.1 Network and Threat Model

Wireless sensor networks generally deployed in open areas. Third parties can determine some attributes of detected events by using their own sensors or by directly observing events. They can perform record linkage attack in order to identify the event owner. Our threat model bases on providing privacy by preventing "record linkage attack" during data collection.

The privacy threats are both due to the eavesdroppers and the sink. That is why our threat model should address the privacy requirements of these two threat types.

In our model, the privacy requirement levels of the system against the sink and against the eavesdropper are not the same. Thus, we employ a privacy mechanism in which there are two privacy levels associated with 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 captured data has to be protected to some extent. In this way, the eavesdroppers can only learn limited information out of the captured data.

On the other hand, *semi-trusted* sink is allowed to legally obtain data, but 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 as compared to the eavesdroppers, but the information detail is still limited to some extent.

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.

In order to prevent "record linkage attack", k-anonymity can be provided by fusing different events. Therefore, trusted entities where anonymizations take place have to be determined. Due to threats of WSN environment, these fuse points have to be distributed so that capturing of one point do not lead to compromising of whole network data. Also it is convenient to choose points as much close as possible to 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 4.1, the basics of the network and threat models are shown. The links between sensor nodes and aggregation nodes are assumed to be secure. Here, in order to provide confi-

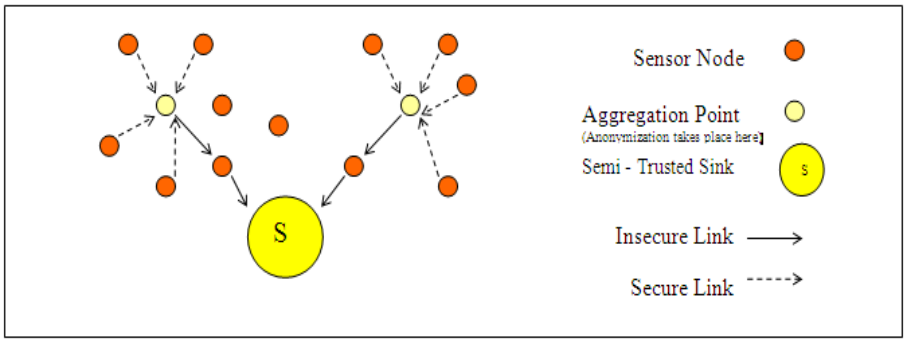


Figure 4.1: Visualization of Network and Threat Models

dentiality of the traffic between sensor nodes and aggregation node, an appropriate key management mechanism [56, 57] could be employed. On the other hand, the links between aggregation nodes and semi-trusted sink are assumed to be insecure. Actually, we address the privacy issues of that part of the network in this chapter.

We assume that our WSN uses widely accepted data-centric routing protocols like SPIN [58] or directed diffusion [59] for finding appropriate routes from sensors to sink.

In our model, it is assumed that one individual generates one event in the anonymization period. In other words, aggregation points process independent events. In this way, we remove the possibility of having correlation among the records that we anonymize.

4.1.2 Two Level Of Privacy with mk-ACM Method

In this section, modification of k-ACM Method that was described in Chapter 3 is given in order to fulfill the requirements of two different levels of anonymity. New version of k-ACM is named as modified k-Anonymization Clustering Method (mk-ACM).

mk-ACM starts with the initialization phase where a new cluster is created

for each input vector as in k-ACM. After the initialization, clustering operation starts and clustering is performed in two distinct stages, $k1$-anonymization and $k2$-anonymization stages. mk-ACM is given in Algorithm 4.2. The notation table given Table 3.1 is also used in this section.

Algorithm 4.1 Function Cluster Combination

Input: parameter, k, distance matrix, D^h
Output: New cluster, L_u^h, updated distance matrix, D^{h+1}
1: Find clusters, L_s^h, L_t^h, having minimum distance in distance matrix D^h
2: create a new cluster L_u^{h+1}
3: $V_u^{h+1} \leftarrow V_s^h \bigcup V_t^h$
4: $|V_u^{h+1}| = |V_s^h| + |V_t^h|$
5: **for** each i^{th} bit string of representative vector **do**
6: $\quad R_u^{h+1}[i] \leftarrow R_s^h[i]$ **or** $R_t^h[i]$
7: **end for**
8: Remove clusters, L_s^h, L_t^h
9: Find the distance of L_u^{h+1} to other clusters, update D^{h+1}

In the $k1$-anonymization stage (9-12. items in Algorithm 4.2), mk-ACM forms the clusters in a bottom-up fashion until each cluster represented has at least $k1$ records. A sample tree structure of clusters obtained at the end of the $k1$-anonymization stage is shown in Figure 4.2. Here, $h1$ is defined as the iteration number needed to complete the $k1$-anonymization stage. 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 (14-17. item in Algorithm 4.2, bottom-up clustering starts with the clusters represented by the root nodes of $k1$-anonymization tree and continues until all the root nodes of tree have sizes of at least $k2$, where $k2 \geq k1$. A tree structure obtained at the end of $k2$-anonymization stage is shown in Figure 4.3. Here, $h2$ is defined as the iteration number after which all the root nodes have at least $k2$ items and bottom-up clustering is completed. This tree structure has $c1$ leave nodes, $L_1^{h1}..L_{c1}^{h1}$ and $c2$ root nodes, $L_1^{h2}..L_{c2}^{h2}$.

Algorithm 4.2 Main Function of mk-ACM

Input: Table, T, number of records, n, number of attributes, m, anonymization parameters $k1$, $k2$ ($k2 \geq k1$), output enlargement factor, M
Output: Anonymized table, mk-ACM(T)

1: $h = 1$ {Initialization}
2: **for** all i where $\{i : 0 < i < n\}$ **do**
3: Create cluster array, $\{L_i^1\}$
4: Add record, T_i to V_i^1
5: Set initial size of cluster, $|V_i^1| = 1$
6: Initialize the representative vector, $R_i^1 \leftarrow T_i$
7: Initialize the distance matrix D^1 by using Equation 3.9
8: **end for**
{$k1$-anonymization}
9: **while** not for each cluster $|V_i^h| \geq k1$ **do**
10: Call Function ClusterCombination ($k1$, D^h)
11: $h = h + 1$
12: **end while**
13: $c1 = |L^h|$, $h1 = h$
{$k2$-anonymization}
14: **while** not for each cluster $|V_i^h| \geq k2$ **do**
15: Call Function ClusterCombination ($k2$, D^h) given in Algorithm 4.1
16: $h = h + 1$
17: **end while**
18: $c2 = |L^h|$, $h2 = h$
19: Initialize the set of vectors to be sent to sink, S, by clusters $L_1^{h2}, ..., L_{c2}^{h2}(|S| = c2)$
20: $\varphi = c2 + M.(c1 - c2)$ (φ is max. number of representative vector allowed for sending)
{Output enlargement for partial encryption}
21: **while** not $|S| = \varphi$ **do**
22: Select the node, f, with the maximum information loss in S
23: Find the child nodes, g and h, of node f
24: Modify S by replacement of node f with nodes g and h
25: **end while**
26: mk-ACM(T)$\leftarrow \varnothing$
{(**Form the output of mk-ACM**)}
27: **for** each cluster, C, in S **do**
28: **if** $|C| \geq k2$ **then**
29: Append representative vector of C and $|C|$ to mk-ACM(T)
30: **else**
31: Append $|C|$ and encrypted version of C to mk-ACM(T)
32: **end if**
33: **end for**

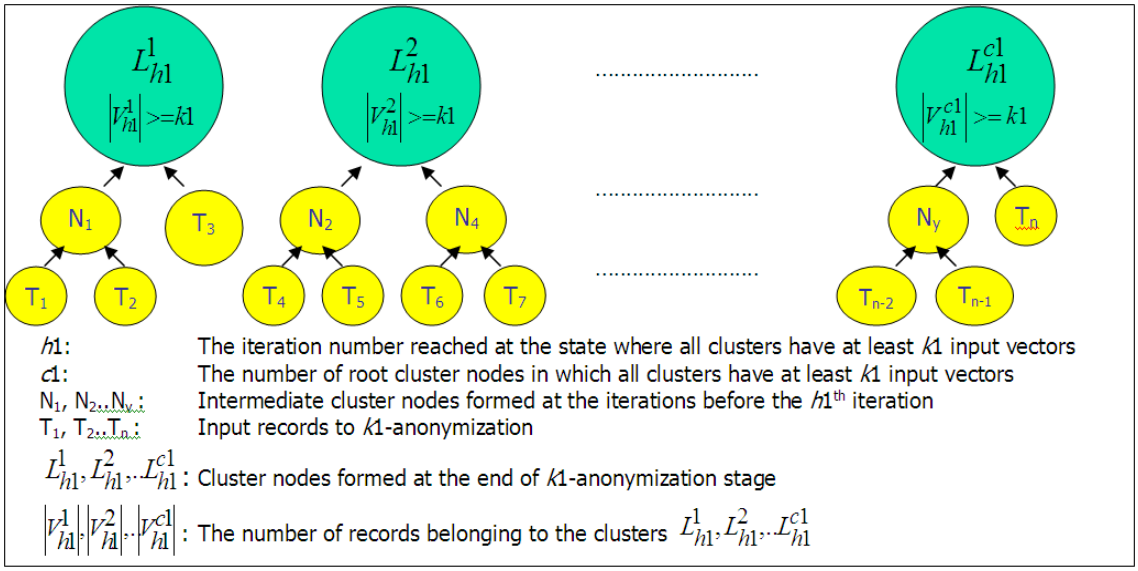


Figure 4.2: A sample tree structure of clusters obtained at the end of $k1$-anonymization stage

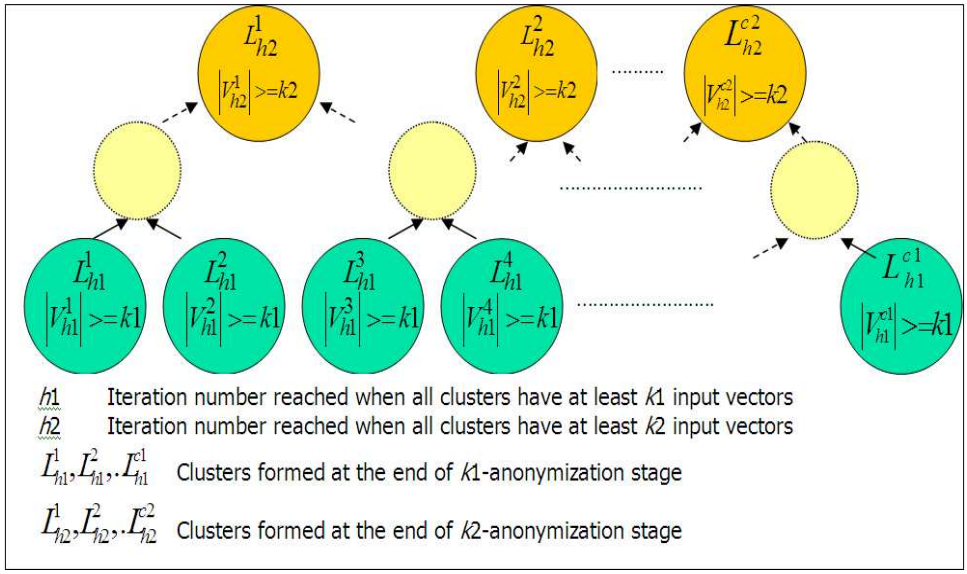


Figure 4.3: A sample tree structure of clusters obtained at the end of $k2$-anonymization stage

Before explaining 21-25. items of mk-ACM method in Algorithm 4.2, the motivation behind output enlargement and partial encryption is given below. As discussed before, information loss is an important design criterion in mk-ACM. Another important criterion is energy saving. Sensor nodes are scattered in an area where there is no power supply other than a simple battery. Therefore, increasing the lifetime of battery 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 encryption process needs a considerable amount of CPU processing, recent studies [55] 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 goes over many hops in the network, energy saving for transmission/reception operations becomes a crucial design criterion. These facts direct the wireless sensor network (WSN) designers to shorten the length of the packets.

Actually, there is a tradeoff between information loss and energy consumption. We analyze this tradeoff in two extreme cases of mk-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 14-17. items of Algorithm 4.2 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 12. item of Algorithm 4.2 are entirely encrypted.

Both cases fulfill the requirements of our threat model. In the first case, at 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. This yields maximized energy consumption. However, since encrypted data will be decrypted at sink, there is no extra information loss here.

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

mk-ACM can effectively find the appropriate data entries for encryption operations (21-25. items of mk-ACM method in Algorithm 4.2) as described below. It firstly $k2$-anonymizes the data (14-17. items of mk-ACM algorithm) as depicted as a tree structure in Figure 4.3. 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 root nodes of the tree, $L_1^{h2}..L_{c2}^{h2}$. All of these nodes are $k2$-anonymized. If no encryptions are allowed, this initial content of S is sent to sink that causes maximum information loss as discussed above. To reduce information loss, the encrypted versions of representative vectors with k-anonymization levels less than $k2$ can be sent to sink. This process requires moving down the tree in Figure 4.3. In other words, some nodes in S are replaced by their children. This is done in an iterative manner until a certain limit. At each iteration, mk-ACM chooses the element of S with highest information loss and this element is replaced by 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 now discarded some generalization operations by moving down

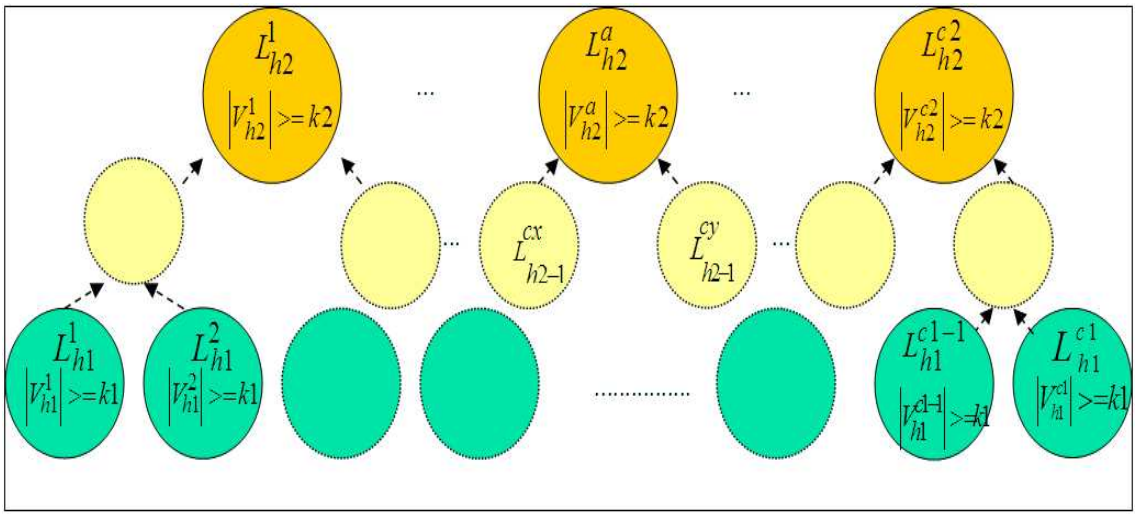


Figure 4.4: Selection of data entries for encryption

the tree.

An example is depicted in Figure 4.4. Assume that node, L_a^{h2}, has the maximum information loss and 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 nodes, $L_1^{h2}..L_{cx}^{h2-1}, L_{cy}^{h2-1}..L_{c2}^{h2}$. By this replacement, we can cancel the information loss that have had occurred after the merging of nodes, $L_{cx}^{h2-1}, L_{cy}^{h2-1}$. Therefore, the quality of data represented by the new set S is greater than the previous one. However, the length of S is increased by one, L_{cx}^{h2-1} and L_{cy}^{h2-1} must be encrypted since they are not $k2$-anonymized.

In 21-25. items of mk-ACM given in Algorithm 4.2, the number of replacements is adjusted by a predetermined threshold value, 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.

After the end of replacement process, vectors of nodes in S are transmitted to the sink together with the node sizes (27-33. items of mk-ACM). Before transmission, representative vectors of nodes having anonymity levels less than $k2$ are encrypted in order to make the data $k2$-anonymous for eavesdroppers. The other vectors, which

are actually $k2$-anonymous, are sent in clear-text.

In mk-ACM, M determines the tradeoff between information loss and energy saving. If M is zero, then $c2$ $k2$-anonymized data entries are sent to 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 more quality data at the cost of higher energy consumption for both communication and encryption. Small M values mean smaller output size and less quality data at the benefit of less energy consumption. The analysis for this tradeoff is given in Section 4.1.4.

Network-wide Operation of mk-ACM

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

The pseudo-code which shows the network-wide operation of mk-ACM in a parallel manner is shown in Algorithm 4.3.

Algorithm 4.3 Network-wide operation of mk-ACM algorithm in a parallel manner

```
1: for each aggregation node in parallel do
2:         while while time period is not exceeded or buffer of node is not full do
3:                 Collect and accumulate local event information from sensors
4:         end while
5:         Run mk-ACM
6:         Send the output of mk-ACM to the sink
7: end for
```

4.1.3 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. Basically, k-anonymization makes the quasi-identifier fields of k or more records identical. It is not needed to resend these identical parts again and again. These parts can be sent to the sink only once along with the number of occurrences. By this reduction technique, the data is shortened as well although information about the individual events is conserved.

In section 4.1.2, it is described that actually the mk-ACM output is composed of representative vectors of clusters and their sizes. The number of representative vectors is determined by the value of $c2+M(c1-c2)$ in which M is the predetermined threshold value, called output enlargement factor. This variable can take values from zero to one. As a result the range for 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 value. $k1$, $k2$ are the parameters of the k-anonymization. The size of clusters cannot exceed $2k_2$, therefore cluster size can be represented in the output with at most $log(2k_2)$bits. 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 data is $mn \log p$. 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 overall k-ACM output as follows:

$$D = \frac{mn \log(p) - (c2 + M(c1 - c2)).(mp + log(2k_2))}{mn \log(p)}. \tag{4.1}$$

Decrease ratio directly affects the energy consumption by saving certain amount of energy due to reduced length of data transferred towards the sink. The energy saving metric, C_G, is defined as the ratio of saved amount of energy due to k-

anonymization. By using the decrease ratio, D, energy saving, C_G, is calculated as follows:

$$C_G = 1 - \frac{2.5h_{sregion} + 2.5(1-D)h_{region} + 8.58\ 10^{-4}\beta}{2.5(h_{sregion} + h_{region})}. \tag{4.2}$$

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 where it is actually $M(c1 - c2)$. Derivations for $h_{sregion}$, h_{region} and C_G are given as follows:

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

$$d_{sregion} = \int_{x=0}^{X_{sregion}} \int_{y=0}^{Y_{sregion}} \sqrt{(x - (X_{sregion}/2))^2 + (y - (Y_{sregion}/2))^2} f(x) f(y) dx dy. \tag{4.3}$$

Here, $f(x)$ and $f(y)$ are the probability distribution functions of 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:

$$h_{sregion} = d_{sregion}/R. \tag{4.4}$$

From sensor node to the aggregation node, an event message travels $h_{sregion}$ hops

which is calculated as follows:

$$d_{sregion} = \int_{x=0}^{X_{sregion}} \int_{y=0}^{Y_{sregion}} \frac{\sqrt{(x-(X_{sregion}/2))^2 + (y-(Y_{sregion}/2))^2}}{X_{sregion} Y_{sregion} R} dxdy. \quad (4.5)$$

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

$$d_{region} = \int_{x=0}^{X_{region}} \int_{y=0}^{Y_{region}} \frac{\sqrt{(x-(X_{region}/2))^2 + (y-(Y_{region}/2))^2}}{X_{region} Y_{region} R} dxdy. \quad (4.6)$$

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 [55]. We assume that the data is processed in Sensoria's WINS NG RF subsystem with MIPS R400 processor where encryption algorithm is AES. The transmission/reception, transmission/encryption and encryption/decryption energy consumption ratios for the same length 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 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

Table 4.1: Energy Consumption Ratios

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

T_D, are 4.29e-4 units.

Total consumed energy without k-anonymization is denoted as C_N. In this case, all event messages are sent to aggregation node, but 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 consumed energy 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:

$$C_N = (h_{region} + h_{sregion})(T_T + T_R)e = (h_{region} + h_{sregion})2.5e. \tag{4.7}$$

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 Moreover, in this case, β bytes of the event message are encrypted. The energy consumption, C_K, can now be calculated as follows:

$$C_K = h_{sregion}e(T_T + T_R) + h_{region}(1 - D)e(T_T + T_R) + \beta(T_E + T_D). \tag{4.8}$$

Total energy saving C_G is calculated as follows:

$$C_G = 1 - \frac{C_K}{C_N} = 1 - \frac{2.5h_{sregion}e + 2.5h_{region}(1-D)e + 8.58\ 10^{-4}\beta}{(h_{region} + h_{sregion})2.5e}. \quad (4.9)$$

In order to calculate the energy saving just after $k1$-anonymization stage of k-ACM algorithm, decrease ratio at this stage, $D_{k1-anonymization}$, have to be calculated. For this calculation Equation 4.1 is needed to be revised. The number of representative vectors at this stage is $c1$ and a cluster size occupies $log(2k_1)$ bit length. $D_{k1-anonymization}$ is found as follows:

$$D_{k1-anonymization} = \frac{mn\log(p) - c1.(mp + \log(2k_1))}{mn\log(p)}. \quad (4.10)$$

If the value of D in Equation 4.2 is replaced with $D_{k1-anonymization}$, then the energy saving that is guaranteed at the end of $k1$-anonymization stage is calculated. Decrease ratio in $k2$-anonymization stage is calculated as follows:

$$D_{k2-anonymization} = \frac{c1.(mp + \log(2k_1)) - (c2 + m(c1-c2)).(mp + \log(2k_2))}{c1.(mp + \log(2k_1))}. \quad (4.11)$$

Energy saving in this stage can be calculated by Equation 4.2 where D is actually $D_{k2-anonymization}$.

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

4.1.4 Performance Evaluation of mk-ACM

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

In the first subsection, the performance of $k1$-anonymization stage is evaluated. Especially, information loss in the data shared with the semi-trusted sink is an important evaluation criterion in this stage. The second subsection evaluates the performance of dynamic taxonomy trees. The third subsection focuses on the performance of mk-ACM in the $k2$-anonymization stage.

Performance Evaluation of $k1$-anonymization Stage

We analyze the performance of $k1$-anonymization stage using the information loss, anonymity level and energy saving metrics. In this way, the performance of bottom-up clustering method as a general k-anonymity solution is investigated. Table 4.2 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 Equation 3.11. This anonymity score is obtained if all clusters have exactly $k1$ elements at the end of mk-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. 'Information Loss' column gives the information loss of $k1$-anonymity operation by using the conditional entropy of Equation 3.5. The last column of Table 4.2, 'Energy Saving', gives the energy saving of mk-ACM in $k1$-anonymization

Table 4.2: Experimental results of data set with 500 records for the $k1$-anonymous stage

$k1$ **value**	**Required Anonymity Level**	**Anonymity Level**	**Information Loss (bits) Value Range:[0.0 2.0]**	**Energy Saving (%)**
3	1.58	2.11	0.54	27
4	2	2.25	0.47	34
5	2.32	2.82	0.77	52
8	3	3.41	0.93	65

stage within the whole energy consumption of WSN due to the reformatting operation proposed in algorithm. In this reformatting operation, the iterated parts in the k-anonymous data is sent once together with the number of occurrences of these iterated parts in the data. In this way, the length of messages decreases. Energy saving at the end of $k1$-anonymization stage is calculated using Equation 4.2 and Equation 4.10. In our analysis, we take the size of WSN field as 500m x 500m, 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. Information loss of 3-anonymous data is 0.54 and the whole system energy saving ratio is 0.27. 5-anonymous data provide an optimal solution such that the information loss value, 0.77, is tolerable and energy saving, 0.52, is quite high. For 8-anonymous data, energy saving is very high (0.65), however, information loss is also high (0.93) that makes the data low quality.

In Figure 4.5, the effect of change in the number of records to the information loss is analyzed for various $k1$ values. For a given $k1$ value, information loss has a general decreasing pattern as the number of records increases. The main reason behind this decrease in information loss is that while the number of records increase, data naturally becomes $k1$-anonymous and fewer generalizations are performed. However, in a few cases (e.g. transition from 500 to 600 records for $k1 = 8$), information loss increases as opposed to general decreasing pattern. These exceptions are due to the nature of the clustering mechanism. The clustering mechanism may occasionally

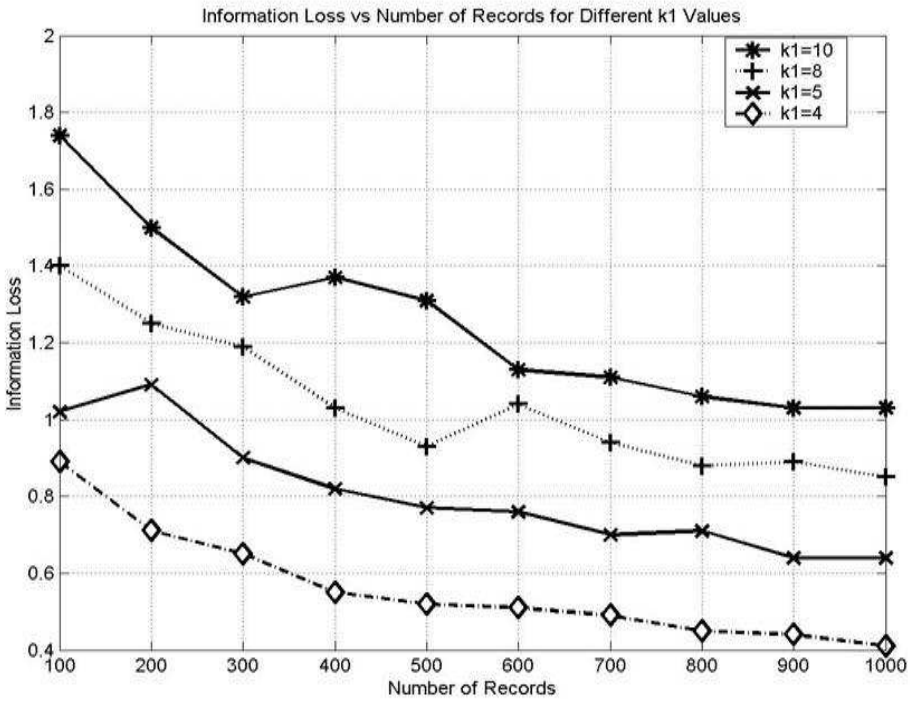


Figure 4.5: Information loss versus record number for different $k1$ values

cause a higher average number of data records per cluster as the number of records increase. This may cause a slight increase in information loss in some cases.

In above experiments, we use data which are generated randomly by using uniform distribution We also performed experiments by using data which are generated by some non-uniform irregular distributions. In terms of information loss, the results of data with non-uniform distributions are better than the ones with uniform distribution in these experiments. If the data is produced by non-uniform distributions, some input vectors are closer to each other. Information loss of those vectors are lower at the end of the anonymization. In other words, by using uniformly distributed data, we presented the worst case performance of our method in Table 4.2 and Figure 4.5. For the sake of simplicity and clarity, we do not give the experimental results obtained by using non-uniformly distributed data.

Performance Analysis of Dynamic vs Static Taxonomy Trees

Notion of dynamic taxonomy tree is proposed in k-ACM and mk-ACM in order to minimize information loss. However, the length of anonymized data produced by using this type of tree is much more than the data generated by static taxonomy tree. A categorical attribute having p distinct attribute values can be represented by p bits according to dynamic taxonomy tree method. If static tree having t total nodes (including leaves and internal nodes) is chosen, the length of data would be $\log(t)$. The value of t can have the maximum value, $2p-1$, when the tree is binary and can have the minimum value, $p+1$, when all leave nodes are directly connected to a root node. Therefore, range of data length is between $[\log(p+1), \log(2p-1)]$. For all $p \geq 2$, static tree data representation is shorter and the length difference gets enormously bigger as p increases.

As stated before, energy consumption of WSN is mostly determined by the length of messages transferred in the network. Actually, there is a trade-off between data utility and energy consumption in choosing the dynamic taxonomy tree method as a data representation method. In this section of chapter, this trade-off is analyzed by experiments.

In the experiments, the number of input records is chosen as 500. Input records have five categorical attributes where each of them is determined as quasi-identifier. Each categorical attribute has six distinct attribute values. It is assumed that the tree given in Figure 4.6 is the static taxonomy tree used for representation of attributes. The tree consists of one root node, two internal nodes and six leave nodes. The length of each attribute is represented by 6 bits in dynamic tree whereas it is represented by $\lceil \log(9) \rceil = 4$ bits in static taxonomy tree. Generalization is actually replacing the attributes with their common ancestor in static tree. In order to be consistent with the other experimental results, information loss metric given in Section 3.2 of Chapter 3 is used. Final information loss is calculated by converting

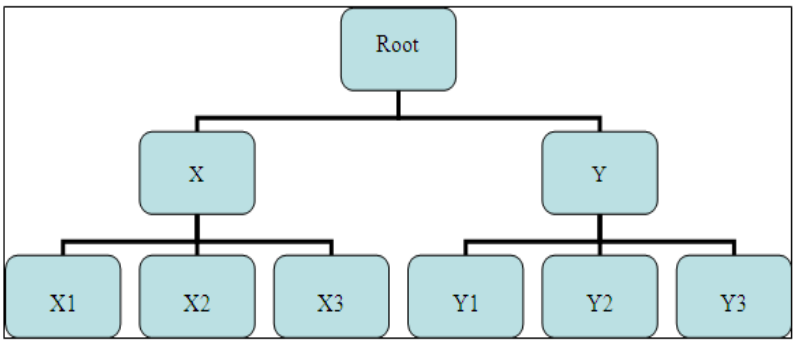


Figure 4.6: A Static Taxonomy Tree Sample

Table 4.3: Experimental results of data set with 500 records for Comparison of Static and Dynamic Taxonomy Trees

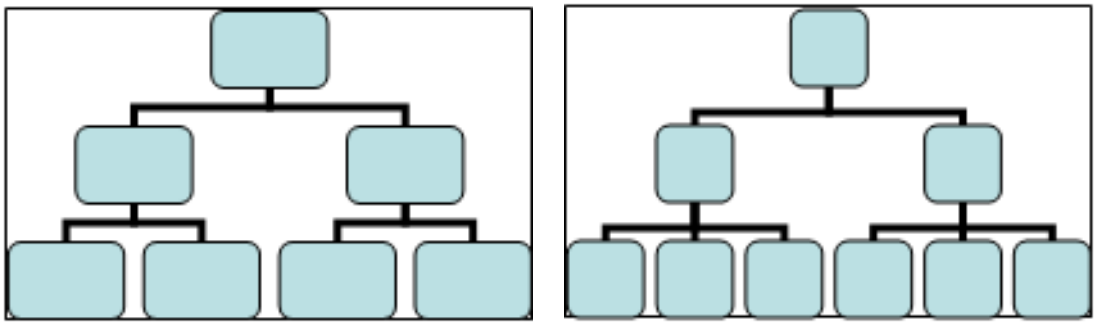
k value	**Info. Loss of Static Taxonomy Tree (bits) Value Range: [0.0 2.58]**	**Energy Gain of Static Taxonomy Tree (%)**	**Info. Loss of Dynamic Taxonomy Tree (bits) Value Range: [0.0 2.58]**	**Energy Gain of Dynamic Taxonomy Tree (%)**
4	1.39	65	0.87	36
5	1.48	69	1.06	48
8	1.95	80	1.48	66
10	1.96	81	1.61	70

categorical attributes of static taxonomy tree into equivalent bit strings at the end. If attribute is generalized into an internal node, all bits corresponding to the children of that node is converted to '1'.

The results of experiments are shown in Table 4.3. Each row corresponds to results of information loss and energy saving values of using static or dynamic taxonomy tree methods for each given k value. Tree given in Figure 4.6 is chosen as a static taxonomy tree. As expected, the information loss caused by dynamic tree is considerable smaller than the loss generated by static tree. For example, in experiment where k is chosen as 5, information loss of dynamic tree is 1.48 bits. However, this value is 1.95 bits in static tree. On the other side, energy saving of static tree method is higher due to the smaller lengths of data representations.

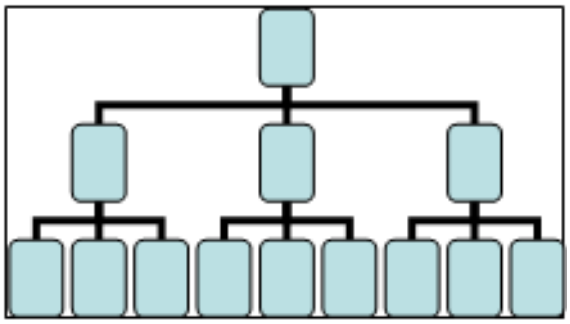
The ratio of energy saving (ES) to the information loss, ES/IL, is determined

for better analysis of trade-off between data utility and energy consumption. The number of input records is 500 each having 5 categorical attributes. Experiments are repeated for different static taxonomy tree structures given in Figure 4.1.4. The numbers of distinct attribute values are 4, 6 and 9 respectively. Ratios, ES/IL, are calculated in each experiment using dynamic and static taxonomy trees as given in Figure 4.8. As a general, in terms of ES/IL, there is no any major difference between dynamic and taxonomy tree methods. Dynamic one maximizes data utility by paying higher costs for energy. If WSN has limited resources for energy and if owners of WSN can tolerate information loss, dynamic taxonomy tree method can be considered as best alternative.



(a) Tree with 4 distinct attribute values

(b) Tree with 6 distinct attribute values



(c) Tree with 9 distinct attribute values

Figure 4.7: Static taxonomy trees for different number of attribute values

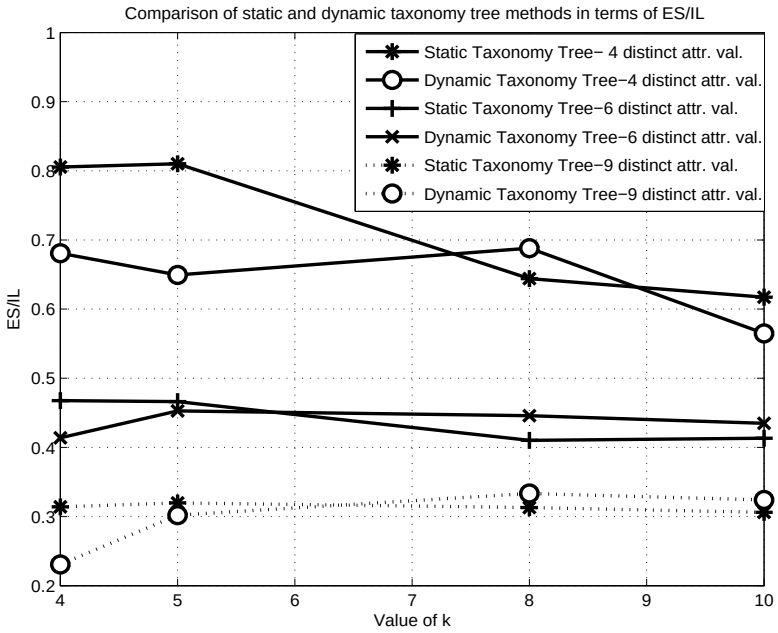


Figure 4.8: Comparison of Static and Dynamic Taxonomy Trees

Performance Evaluation of $k2$-anonymization Stage

$k2$-anonymization stage starts with the $k1$-anonymous data obtained from the previous stage. Output enlargement factor, M, adjusts the length of 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 4.1.2. Increase in the number of encrypted data entries creates more accurate output and decreases information loss.

Table 4.4, gives the experimental results for a data set having 500 records where $k1 = 4$ and $k2 = 16$. First column gives the value of output enlargement factor, M, that is pre-determined for the corresponding experiment. The second column shows the anonymity level that the output provides and it is computed according to Equation 3.12. This anonymity level is obtained after the encrypted parts are decrypted at the sink. Third column gives the information loss occurred only in $k2$-anonymization

stage.It is actually the difference between information loss occurred at the end of $k2$-anonymization stage and information loss at the end of $k1$-anonymization stage. Since information loss of 4-anonymous data is 0.47 and information loss can be at most 2.0, information loss that can occur during $k2$-anonymization can be found as at most 1.53.

Equation 3.5 is used for computations of information losses. Energy saving values in $k2$-anonymization stage are given in the fourth column. Decrease ratio of this stage is calculated by Equation 4.11 and energy saving is found by Equation 4.2. Energy saving values indicate the energy savings as compared to the extreme case where the entire $k1$-anonymous data is encrypted. Last row actually shows this extreme case. Here, $M = 1$, which means output length is maximum and all the data is encrypted. Therefore, information loss and energy saving are zero. The first row shows the other extreme case where the data is $k2$-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 enough. In this case, the quality of data is very low but energy saving 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.

Information loss and energy saving trade-off can easily observed in Table 4.4; as the information loss decreases energy saving also decreases. The designer of WSN can decide on the value of M according to the experimental results about information loss and energy saving, and using its system parameters and requirements. If the system can tolerate more information loss, it is possible to save considerable amount of 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 4.9 by

Table 4.4: Experimental results of data set with 500 records for $k2$-anonymization part

Output Enlargement Ratio	**Anonymity Level**	**Information Loss (bits) Value Range:[0.0 1.53]**	**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

using different M values and different $k1$, $k2$ pairs where number of records is 500. As expected, increase in M decreases data loss. Again as expected, high $k2$ values cause higher information loss as compared to low $k2$ values. However, this difference becomes marginal as M increases. Moreover Figure 4.9 also shows that, especially for high $k2$ values, k-ACM reduces the information loss quickly for small M values.

Figure 4.10 shows the change of energy saving with respect to output enlargement factor for various $(k1, k2)$ pairs. As expected, higher M values result in lower energy saving. As in the information loss case shown in Figure 4.9, high $k2$ values cause more energy saving, but this advantage becomes marginal as M increases. However, differently from the information loss decrease shown in Figure 4.9, the decrease in energy saving is linear.

The purpose of mk-ACM is maximization of energy saving and minimization of information loss. This tradeoff is managed via M. In order to do a better analysis of this tradeoff in $k2$-anonymization stage, we analyze the ratio of energy saving to the information loss (ES/IL) for networks with different $k1$ and $k2$ values. These results are depicted in Figure 4.11. Higher ES/IL ratio shows the effectiveness of output enlargement factor at this stage. It is observed that ES/IL ratio constantly increases as M increases. This fact constitutes another proof for the effectiveness of mk-ACM algorithm. Although ES/IL has a constant increasing pattern, this

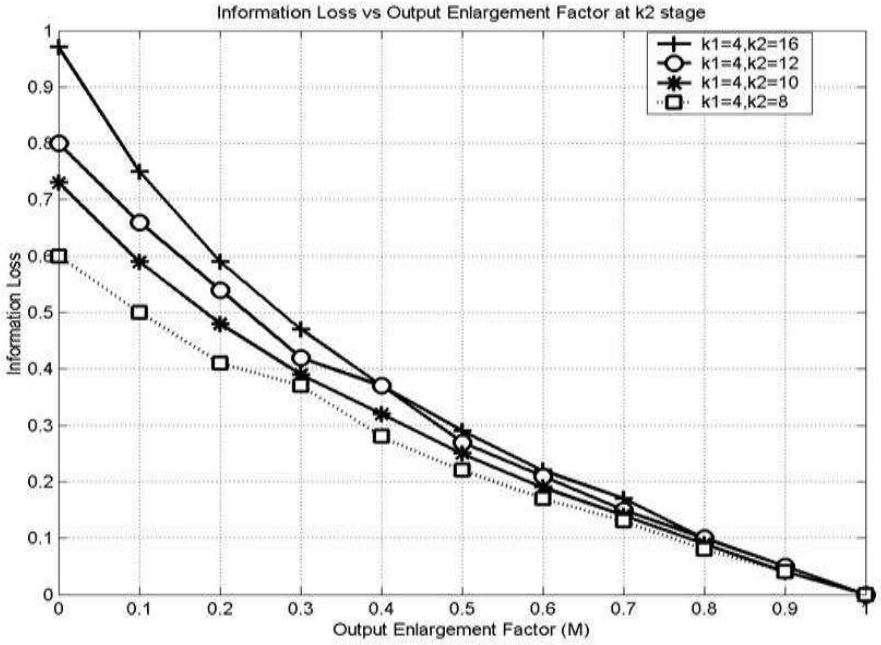


Figure 4.9: Output enlargement factor vs. Information Loss at the $k2$-Anonymization Stage

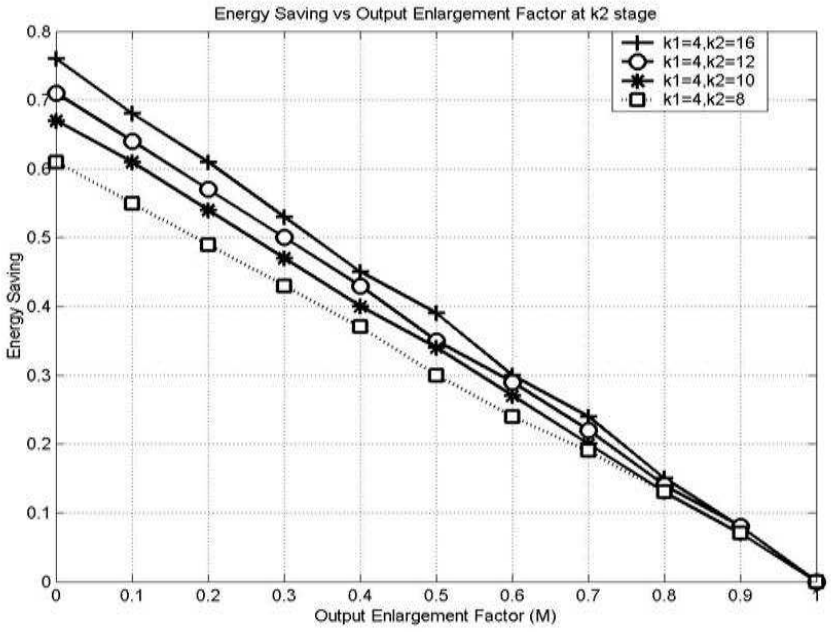


Figure 4.10: Output enlargement factor vs. Energy Saving at the $k2$-Anonymization Stage

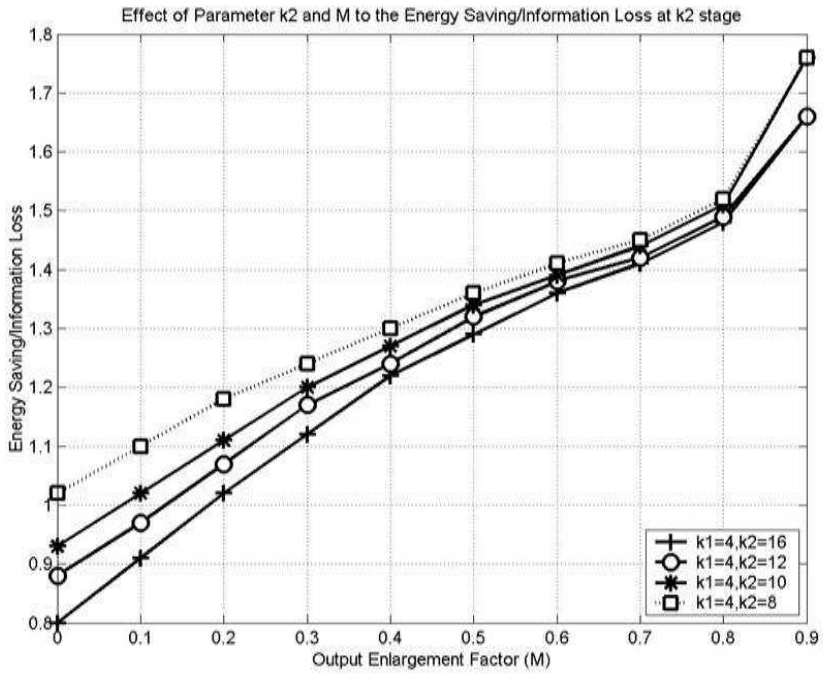


Figure 4.11: Energy Saving/Information Loss for Different M values at the $k2$-Anonymization Stage

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 considerable energy as shown in Figure 4.10. Actually, the network administrators should choose the largest M value that gives the required information loss and/or energy saving values. In this way, the tradeoff between information loss and energy saving can be dynamically managed by playing with M. For example, consider a network with $k1 = 4$ and $k2 = 10$. If the maximum information loss that this network can tolerate is 0.40, then Figure 4.9 suggests to use the $M = 0.3$. In this situation, the network provides the required anonymity levels by saving 47% energy as shown in Figure 4.10. On the other hand for the same network, if the limitation is to save at least 60% of energy, then M is chosen as 0.10. In this case, information loss becomes 0.59 for the same anonymity levels.

4.2 Multiple Sinks

In this section, k-ACM method is adapted as a privacy framework for WSN applications having multiple sinks. Collected event information is iteratively k-anonymized for all sinks each having different privacy levels. Encryption operation with appropriate key management schema is used in addition to generalization in order to meet the different requirements in one k-anonymized output. Achieving all privacy requirements in one output considerably decreases the energy consumption so that this output can be multicasted to multiple sinks instead of sending different outputs for each sink. Bottom-up clustering idea is used during k-anonymization process.

4.2.1 Network and Threat Model

Our threat model bases on the requirement that the individuals do not want sinks to identify their records among other records of k individuals within a specified time-frame through the quasi-identifier fields of records.

The required privacy levels of each sink differs so that suppose that there are n sinks, each i^{th} sink has a privacy level p_i where each level requires to share k_i-anonymous data with i^{th} sink and inequality of $k_1 < k_2 < ... < k_n$ is valid.

Sensors are clustered in separate sensor groups according to sensor localizations where each group has a group head sensor. In our method, each sensor conveys its readings to group heads, they k-anonymizes data and multicast the anonymized output to all sinks when multicasting has advantage in terms of energy saving. Network model is shown in Figure 4.12.

In our model, it is assumed that one individual generates one event in the anonymization period. In other words, group head sensors process independent events. In this way, we remove the possibility of having correlation among the records that we anonymize.

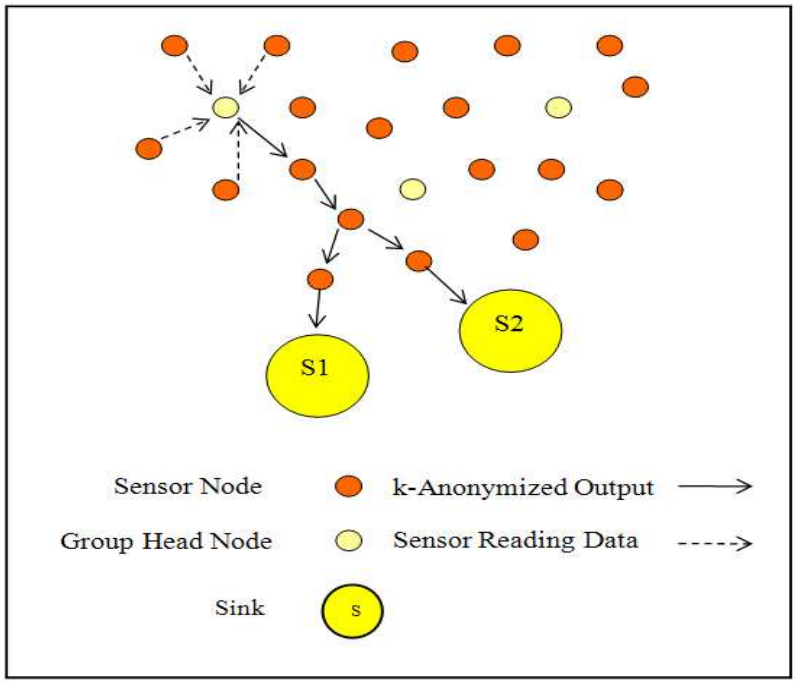


Figure 4.12: Network Model

4.2.2 Iterative k-ACM (Ik-ACM)

In the WSN, there are n sinks and $n-1$ symmetric encryption keys which are labelled as $e_1, e_2, ..., e_{n-1}$. i^{th} sink contains list of the keys as $e_i, e_{i+1}, ..., e_{n-1}$. Each group head sensors store all the $n-1$ keys. In the proposed method, anonymization is completed in n iterative steps as shown Figure 4.13. In the first step, by using only generalization operation, input data is $k1$-anonymized. In the second step, $k1$-anonymized data is $k2$-anonymized by encrypting the chosen data parts with e_1. For each i^{th} step to n^{th} step, anonymization is done by encryption using key, e_{i-1}. The output after n^{th} step is multicasted to all sinks.

After the arrival of anonymous data to sink, each sink decrypts the data with their keys. The resulting data after decryption actually has the level of privacy required for that sink. i^{th} sink can only decrypt the data which is encrypted after the i^{th} iterations; because it has the corresponding keys. Data parts encrypted by

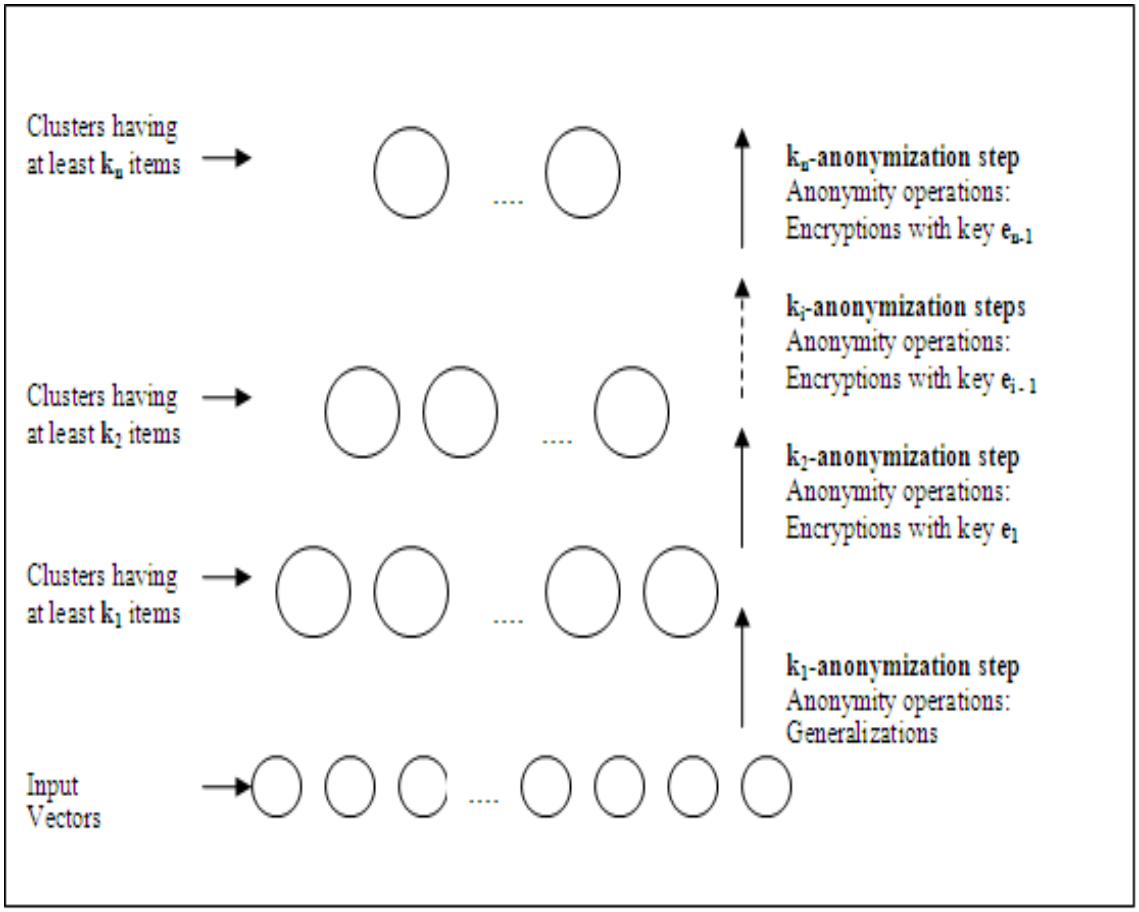


Figure 4.13: Steps of Iterative Anonymization

the keys, $e_1, e_2, ..., e_{i-2}$, cannot be decrypted, therefore they can be considered as suppression operations for that sink. 1^{st} sink, which has to get data with lowest privacy criteria, can decrypt all the encrypted parts and the result data is actually k_1-anonymized. On the other hand, n^{th} sink has not any key and gathers data as kn-anonymized.

Encryption is used as an anonymity operation in all anonymization steps except $k1$-anonymization. Assume that two clusters, L_h^s, L_h^t are chosen as the closest cluster pair at h^{th} iteration of Ik-ACM and this iteration corresponds to $(i+1)^{\text{th}}$-anonymization step. The newly created cluster is labelled as L_h^{st} and $E_{e_i}(x)$ represents the encrypted output of input x with key e_i. Formation of representative vector for newly created cluster, R_h^{st}, is given in Algorithm 4.4.

Algorithm 4.4 Forming Representative Vector in Ik-ACM

Input: Representative vectors, R_h^t, R_h^t
Output: Representative vector of new cluster, R_h^{st}

```
for each attribute m do
    if R_h^s[m] = R_h^t[m] then
        R_h^st[m] = R_h^t[m]
    else
        R_h^st[m] = E_{e_i}(R_h^s[m]||R_h^t[m])
    end if
end for
```

A sample cluster combination by encryption operation is shown in Figure 4.14. Assume that represented vectors of two closest clusters, L_t^h and L_s^h are '0011 0010 1000' and '0100 0010 1000', respectively. The values of vectors indicate that data records have three attributes each having four distinct attribute values. First bit strings of vectors, '0011' and '0100' are compared. Since they are not identical, concatination of values are encrypted. This encryption result forms the first bit string of representative vector of newly created cluster, L_{ts}^{h+1}. Second bit strings of

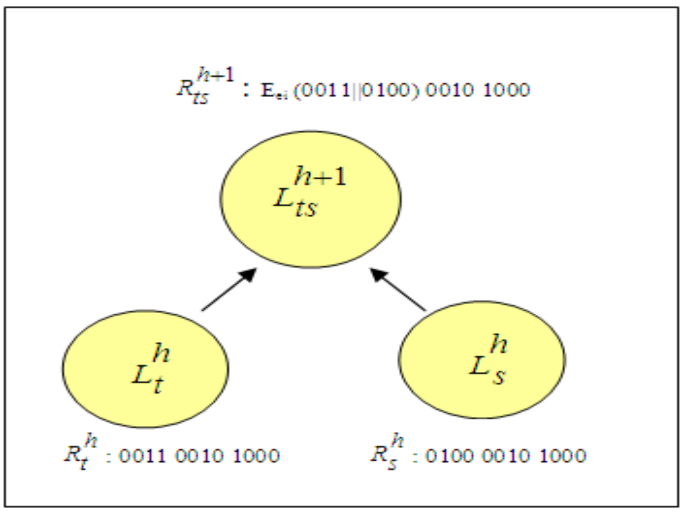


Figure 4.14: A sample case for cluster combination with encryption operations

clusters, '0010' and '0100', are identical. Therefore, second bit string of new cluster is also '0100'. By using the same method, third bit string is found as '1000'.

4.2.3 Multicasting and Energy Gain

Main aim of k-Anonymity solutions is providing the required privacy level with minimum information loss. However another factor, minimization of energy consumption, is an important criteria in WSNs. A sensor node consumes energy for different processes like event sensing, CPU processing, or transmitting/receiving data packets. Among these processes, transmission/reception operations consumes much of the energy. Table 4.1 shows energy consumption rates for transmission/reception which are published in technical report written by Carman et al. [55]. It is stated that energy consumption ratio of transmission/reception to energy consumption of encryption is 2333.34. Since each sensor node acts as a router for the messages of other nodes and one message goes over many hops in the network, energy saving for transmission/reception operations becomes a crucial design criterion. Shortening the length of messages and decreasing the number of travelled hops would help to

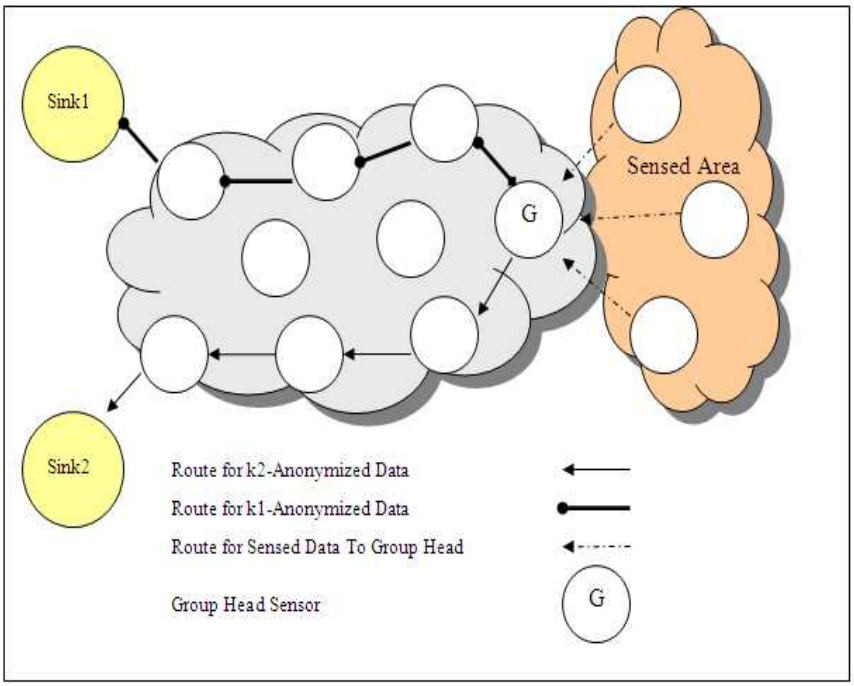


Figure 4.15: Routes when multiple k-anonymized outputs are generated

reduce energy enormously.

In a WSN topology where there are multiple sinks and each sink has different privacy criteria, the basic solution of anonymization is that group head sensor anonymizes the data, produces different outputs for each requirement of sink and sends each output to related sink in different paths as shown in Figure 4.15 (in this figure, there are two sinks in WSN). This sending method is called as *multipath*. However, Ik-ACM produces unique output which is ready for multicasting. One anonymized output that guarantees all the privacy requirements, is sent to a multicast point. After reaching to multicast point, one copy of data is sent to sink1 and the other copy is sent to sink2 as presented in Figure 4.16. Multicasting schema decreases the number of travelled hops for some group head nodes.

Assume that there are two sinks named $Sink1$, $Sink2$. $Sink1$ requires $k1$-anonymized data and $Sink2$ requires $k2$-anonymized data. The lengths of anonymized

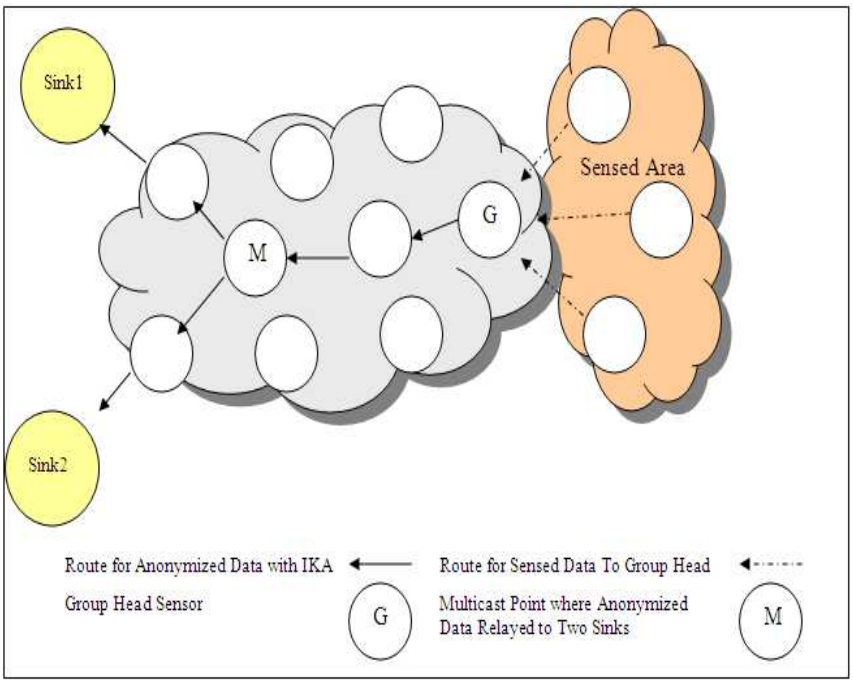


Figure 4.16: Routes when IKA anonymized output is multicasted to sinks

outputs are l_{k1} and l_{k2}, which are obtained by applying directly k-ACM for $k1$-anonymization and $k2$-anonymization, respectively. Data length of anonymous data generated by Ik-ACM is labelled as l_{IKA}. The number of hops in the shortest route from group head sensor, G, to Sink1 and Sink2 is represented as $h_{G,Sink1}$, $h_{G,Sink2}$ respectively. Also assume that the hop distance between G and multicast point, M, is $h_{G,M}$, distances from M to Sink1 and Sink2 are $h_{M,Sink1}$ and $h_{M,Sink2}$, respectively. Unique anonymized data is sent to, M, if an appropriate node exists in the network which holds the Inequality 4.12. Among the possible node candidates, the one which minimizes the value of $h_{G,M} + h_{M,Sink1} + h_{M,Sink2}$ is chosen.

$$(h_{G,Sink1}.l_{k1}) + (h_{G,Sink2}.l_{k2}) > (h_{G,M} + h_{M,Sink1} + h_{M,Sink2}).l_{IKA}. \tag{4.12}$$

If there is no appropriate M point, group head nodes prepare output for each sink and send them in different paths to sinks. Location of sinks and location of

group head sensor effect the selection among multicasting and multipathing.

Inequality 4.12 determines the level of energy consumption of multicasting and multipathing methods since it includes number of hops and the length of messages. It is assumed that the location of sinks are known by group head nodes and they are able to calculate the above inequalities with the routing information they have. Also they form the outputs for multicasting and multipathing, calculate the lengths of outputs, determine and use the best one in terms of energy saving. The first step of Ik-ACM is actually uses k-ACM for $k1$-anonymization. Therefore, extra processing load for calculating the inequality is second $k2$-anonymization step of IkACM for nodes which use multipathing. Nodes in which Multicasting is more appropriate do extra processing for $k2$-anonymization with k-ACM. Since, energy consumption of WSN is mostly determined by message lengths, these extra works do not effect the overall consumption.

4.2.4 Performance Evaluation of Ik-ACM

In this part, information loss and energy gain trade-off is deeply investigated. Since effectiveness of multicasting depends on the locations of sinks and distribution of sensor nodes, different experiments with different WSN topologies are performed.

First experiment investigates the effect of sink locations. The size of WSN field is set to 500m $\times$ 500m. Distance of each hop, R, is taken as 10m. There are two sinks in the field, $sink1$ and $sink2$. It is assumed that each sensor is uniformly distributed through the region. In every region having size of 50m $\times$ 50m, there is one group head sensor node. All the sensors convey their readings to their group head nodes. Then, they anonymize data and relays it to the sinks. Each group head node calculates the cost of multipathing and multicasting for a given set of data, chooses the best method for data sending.

The number of hops from sensor node to group head node and from group head

node to sinks are calculated and they are taken into consideration in calculation of energy consumption. Energy consumption does not only depend on the number of hops; length of the messages are also important for the final results. Message lengths are taken into account during energy calculations.

Assume that energy consumption of method named as "multipath method" is denoted as $E_{multipath}$. Energy consumption of method that uses multicasting when appropriate is represented as E_{hybrid}. Energy gain ratio, EG, is computed as follows:

$$EG = 1 - \frac{E_{hybrid}}{E_{multipath}}. \tag{4.13}$$

Values of $k1$, $k2$ are chosen as 3 and 6 respectively. If all group head nodes use multipath method, information loss for the data sent to $sink1$ is found as 0.44 and data loss for $sink2$ is 0.88. If multicasting with Ik-ACM is used when appropriate, information loss for $sink1$ also does not change since the first step of Ik-ACM is a normal $k1$-anonymization step. However, information loss of $sink2$ and energy gain ratios change. Table 4.5 gives the results obtained for different sink locations. Five different locations as given in the first column are chosen. Second column presents information loss for $sink2$. The number of nodes using multicasting and multipathing methods are shown in third and fourth columns. Last column gives the result of energy gain according to the case when all nodes are using multipathing method.

As the sinks get closer to each other, it is observed that the number of group head nodes using multicasting increase due to finding optimum multicast points. In the case where the sinks are located in coordinates of (0,0) and (500,500), sinks have the maximum distance between each other. 716 group head nodes, out of 2500 nodes, choose multicasting as the best alternative. Information loss increases to 1.03 bits but energy gain is very limited, 3%. In this topology, the number of multicasting nodes is lower and energy consumptions of multicasting and multipathing methods

Table 4.5: Results of using Multicasting and Multipathing together with different sink locations

Location of Sinks (coordinates)	**Info. Loss For Sink2 (bits) Value Range:** [0.0 2.0]	**No. of Group Head Nodes Using Multicasting Method**	**No. of Group Head Nodes Using Multipathing Method**	**EG** (%)
(0,0),(500,500)	1.03	716	1784	3
(0,0),(500,0)	1.14	1287	1213	6
(100,0),(400,0)	1.24	1813	687	14
(150,0),(350,0)	1.31	2137	363	22
(200,0),(300,0)	1.36	2406	94	32

do not differ. If sinks are located in coordinates (200,0) and (300,0), 2406 nodes choose multicasting. In this case, energy gain increases to 32% and information loss increases to 1.36. Actually, there is a trade-off between data utility and energy consumption. If there is a need for energy minimization and *sink*2 can tolerate additional data losses, using multicasting method in some network topologies may be an efficient solution.

Other than the location of sinks, another parameter that may effect the efficiency of multicasting method is the total size of WSN field. Figure 4.17 shows the overall performance results of WSN fields having different sizes and different sink locations. As the size of sensor field increases, multicasting method makes the network consume less energy. For example, when the sink coordinates are (0,0) and (0,500), energy gain is 0.22 for WSN size, 500m×500m, however gain rises to 0.35 for mode wider size, 1000m×1000m. Since, the distance between each sink is not different in both situations, ratio of group head nodes which choose multicasting increases in wider WSN fields. In field size 500m×500m, 1287 of 2500 group head nodes choose multicasting. On the other side, in field having sizes 1000m×1000m, 8787 of 10000 group head nodes select multicasting option. The cost of increasing the energy gain is the lower data utility since information loss increases to 1.32 from 1.14 in 1000m×1000m sized field. In experiments having sink locations like (0,0),

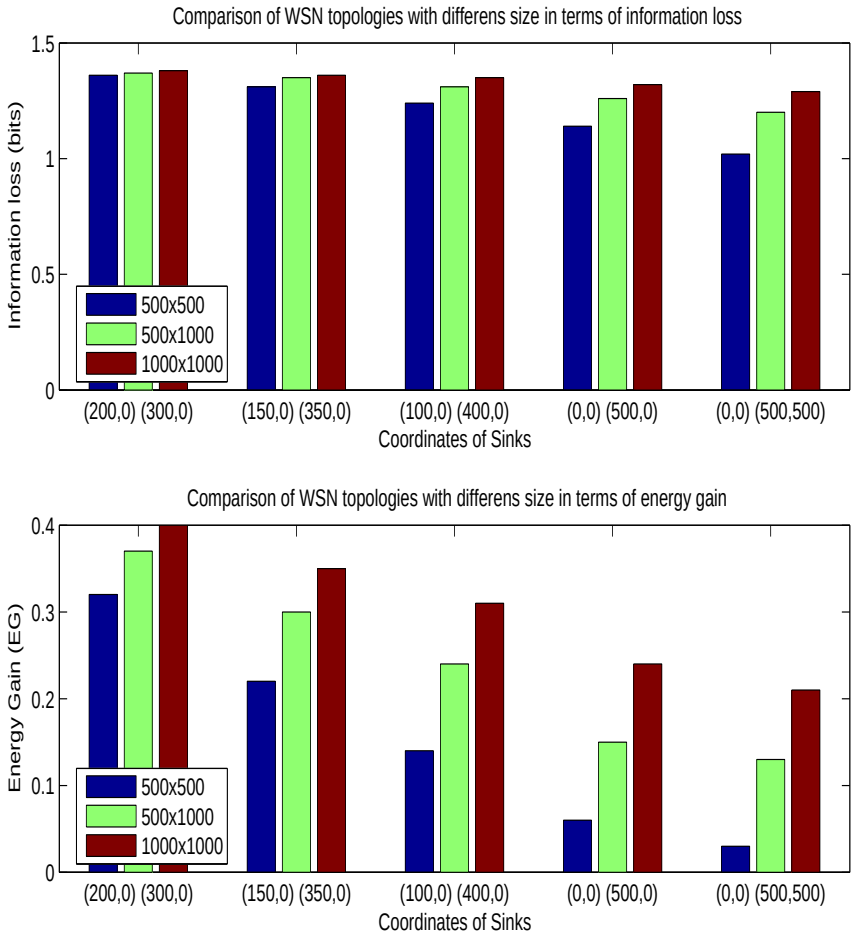


Figure 4.17: Performance Comparison of Topologies with Different WSN area sizes

(0,500) and (0,0), (500,500), energy gains are nearly 3-6% in 500m×500m sized fields which seem that multicasting does not create any promising results. However, in wider sized area like 1000m×1000m, multicasting method yields good results so that energy gain is above than 20%,

Chapter 5

l-Diversity based Framework for Preserving Organizational Privacy in Intrusion Log Sharing Applications

It is known that black-hat hackers share information with each others to choose and attack victims. In underground communities, zero day vulnerability information, target victim information, stolen credit card numbers, bots, spam mail lists, attack tools, etc. are shared or sold easily. On the other hand, system managers, who try to defend their systems against black-hat hackers, need to share related materials about defensive tools, methods and information. In this defensive context, one of the important source of information is logs of intrusion detection systems. Defensive experience of an organization can be easily transferred to others by sharing these logs.

It is common that most of the organizations somehow use intrusion detection systems to detect attacks to their systems. These systems do not always produce useful outputs. Especially the elimination of false positive alarms needs a labor intensive work. An information security expert has to choose correct set of attack signatures which are appropriate for his system and eliminate false positive alarms. However, most of the organizations cannot reserve man power for this task due to lack of specialized technical person or due to lack of budget. Under these circumstances, outsourcing of intrusion log analysis could be a good alternative.

Nowadays, National Computer Emergency Response Teams (NCERT) are try-

ing to find ways for performing proactive nationwide or sectorwide security countermeasures in order to detect and prevent cyber attacks to the national critical information infrastructures. These infrastructures generally belong to different organizations. NCERTs try to find ways to probe them centrally. Probing aims to deduce the overall threat state of each organization and determine the overall threat level of country. For this aim, a distributed intrusion detection system has to be setup and managed. Moreover, collected intrusion logs has to be centrally stored and analyzed.

In both cases, outsourcing of intrusion log analysis of an organization or probing of critical information infrastructures by NCERTs, there is a need to a central intrusion log management office (CILMO) which stores logs of different organizations centrally, analyzes them, detects attacks, send alarms to organizations and generates statistics for determination of organizationwide or nationwide threat levels.

Main obstacle for forming a CILMO is privacy concerns of organizations. Intrusion logs contain valuable information about organizations, like detailed knowledge of targeted information assets, attack times, types of attacks, results of attacks, etc. Organizations are reluctant to share intrusion logs because of two main reasons. First one is that they do not fully trust the personnel of CILMO, because administrators of CILMO may intentionally misuse their attack information. The second reason may be the lack of appropriate security and privacy countermeasures which have to be applied to the intrusion logs during their transmittion, processing and storage. Without solving these security and privacy problems, organizations generally do not want to send their intrusion logs to a CILMO although it may be set up by a NCERT team.

Organizations confront with a dilemma between privacy risks and benefits of sharing intrusion logs. Actually, privacy problem can be solved by hiding private parts of information. In a practical solution, hiding and sharing, which is contra-

dictory in nature to each other, have to be performed at the same time. As the privacy criterion of an organization increases, information loss of sharing operation increases at the same time. Therefore the solution of this problem has to deal with the trade-off between privacy and information loss and this trade-off have to be adjusted according to the needs of organizations.

In this chapter, a privacy preserving framework is presented for intrusion log sharing. This framework is based on l-diversity notion. This notion guarantees that anyone cannot identify the exact sensitive attribute value of an individual among other $l-1$ sensitive attribute values. In our case, sensitive attribute is classification type of an intrusion log. Therefore, l-diversity provides the privacy of intrusion logs shared with CILMO so that any administrator cannot identify the exact classification type of an intrusion belonging to information assets of a specific organization among $l-1$ different sensitive attribute values. Also privacy schema enables us to hide the originator organization of intrusion log among $l-1$ organizations.

By collection of privacy preserved intrusion logs, this framework enables CILMOs to perform detailed security analysis of organizations, draw conclusion about the general security status of organization categories and prepare a warning mechanism.

5.1 Threat and Network Model

In our study, organizations send their intrusion logs to a trusted party. In a realistic scenario, trusted party may be an internet service provider (ISP) or a proxy application, which is served by people trusted by organizations. A sample system topology for proposed privacy framework is given in Figure 5.1. Trusted party anonymizes intrusion logs, strips off the destination IP information of log and appends a destination tag instead of destination IP that represents only the originator organization.

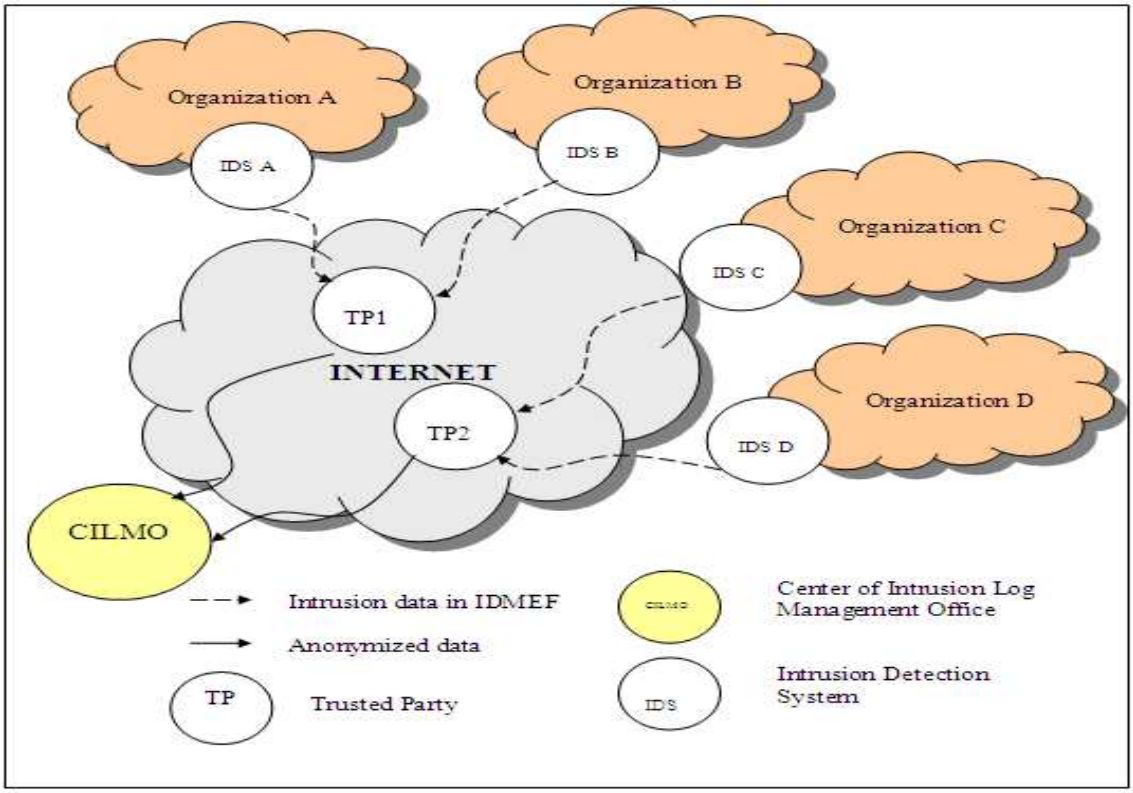


Figure 5.1: System Topology For Privacy Framework

Attributes of intrusion log are classified as shown in Table 5.1. According to this attribute classification, our anonymization method provides prevention of record and attribute disclosure. Therefore, it is needed to provide l-diversity property of intrusion logs.

It is assumed that in each log coming from organizations, pre-exploitation, exploitation and post-exploitation activities are correlated and one log entry is created for each attack. If one attack targets many servers of an organization, only one log entry is produced by IDS. According to these assumptions nobody can bypass the anonymization schema.

Table 5.1: Classification of Intrusion Log Attributes

Attribute	Classification
Target Organization	Identifier
Target Service	Quasi-identifier
Source	Quasi-identifier
DetectTime	Quasi-identifier
Classification	Sensitive

5.2 *l*-ACM for Intrusion Logs

k-ACM method proposed in Chapter 3 is modified for anonymization of intrusion logs in order to have l-diversity property. Proposed method is named as l-Diversity Anonymous Clustering Method (l-ACM).

The stopping criteria for clustering iterations is the major modification issue. k-ACM continues to clustering iterations until all cluster has at least k records. However, l-ACM keeps on iterations up to the point where each cluster contains a record set having distinct l sensitive attribute values.

Numerical attributes are converted to categorical attributes by dividing numeric range of attribute into intervals and representing each interval by a categorical attribute value in k-ACM.

Destination port, source IP and time attributes are also numerical attributes. Converting them to numeric intervals does not yield valuable information for security analysis tasks performed at CILMO side. Exact destination ports or source IP addresses are needed in order to have statistics of port usage or to list the IP addresses of top attackers, respectively. Also exact time information may be used for better security analysis. Since minimization of energy consumption and shortening anonymous output is not a requirement in an intrusion log sharing application, l-ACM does not convert numerical attributes to intervals and it does not represent attributes as bit string. It sends them to CILMO as a set of numerical attribute

values. By using this method, l-ACM does not change the original data. It only disturbs the mappings between different attributes which actually belong to the same record.

Target organization can be considered as identifier of an intrusion log. Normally, privacy preserved data publishing techniques strips off identification information and anonymizes the quasi-identifiers. However, in our case, CILMO needs the names of the target organizations in order to perform the required security analysis tasks. Names of the target organizations are transferred to CILMO in such a way that nobody can deduce the name of exact organization of an intrusion log among $l-1$ organizations.

The same organization may send many intrusion logs to CILMO. If one anonymity set produced by l-ACM has many intrusion logs of the same organization, this situation may violate l-diversity property. Therefore, l-ACM guarantees that each record in each cluster has to belong different organization.

Suppose that, i^{th} cluster gathered at the end of l-ACM has a representative vector, R_i which is actually the anonymous output vector of all quasi-identifier attributes. If the number of quasi-identifier attributes is assumed to be p, vector R_i has the size value of p. Since each quasi-identifier attribute is converted to a set of attribute values, each item consists a set of attribute values.

Suppose that n is the number of records. Set of all target organizations is represented as $\{O_1, O_2, ..., O_n\}$. Assume that all records has m different sensitive attribute values where $m > l$ and these attributes values are $\{S_1, S_2, ..., S_m\}$. The data sent to CILMO can be shown as $\{O_1, O_2...O_n\}, R_i, \{S_1, S_2...S_m\}$.

A running example of l-ACM is shown in Table 5.2 and Table 5.3. Assume that trusted party knows the IP range of each organization and each destination IP belongs to a different organization. Destination IP of intrusion log is replaced with the name of organization during anonymization. Trusted party gathers the

Table 5.2: An Example About Anonymization of Intrusion Logs - Original Data

Dst IP	Src IP	Time	Dst Srv	Classification
201.2.1.10	195.100.4.4	11:00	53	DNS Zone Transfer
223.23.5.4	195.100.4.4	11:30	8080	WEB IIS ISAPI
212.125.12.12	198.166.3.3	11:40	3372	DoS MSDTC
222.19.1.103	190.67.30.3	11:45	1543	NETBIOS SMB
208.234.3.105	199.201.45.56	11:55	80	WEB-COLDFUSION
200.188.5.17	191.34.32.1	12:05	1548	DOS IGMP

Table 5.3: An Example About Anonymization of Intrusion Logs - 2-Diverse Data

Destination Organization	Source IP	Time	Destination Service	Classification
$\{O_1, O_2\}$	{195.100.4.4}	{11:00,11:30}	{8080, 53}	{DNS Zone Transfer, WEB IIS ISAPI}
$\{O_3, O_4\}$	{198.166.3.3, 190.67.30.3}	{11:40, 11:45}	{1543, 3372}	{DoS MSDTC, NETBIOS SMB}
$\{O_5, O_6\}$	{199.201.45.56, 191.34.32.1}	{11:55, 12:05}	{80, 1548}	{WEB-COLDFUSION, DOS IGMP}

original data shown in Table 5.2, produces three clusters each having two elements and makes the data 2-diverse. Each row in this table represents one cluster. All the attributes are converted to sets of distinct attribute values. l-ACM guarantees that in destination organization attribute, two distinct organization names exist and classification attribute consists of a set having two different classification values. Since source IP, time and destination port attributes are chosen quasi-identifiers, l-ACM tries to minimize the number of distinct attribute values of these attributes in anonymized output.

5.3 Warning Mechanism

CILMO may need to warn organizations about a very critical intrusion. Generally, organizations set up intrusion detection systems (IDS) for monitoring of intrusions. However, due to lack of appropriate technical expertise on evaluating IDS logs, these systems are not used properly. CILMO can help organizations by providing technical expertise in analyzing intrusion logs centrally.

On the other side, if the proposed anonymization method is used in intrusion log sharing, CILMO does not know the exact intrusion classification for exact originator. It only knows that a set of organization corresponds to a set of intrusion classification values. CILMO may be interested in one intrusion classification among these values. If it is assumed that trusted party does not store any information including the mappings of original data with anonymous data, the warning can be performed by only distributing it to each IDS management server of all candidate organizations. IDS management server stores all the IDS logs of corresponding organization in a database.

Details of warning mechanism is described with an example in Figure 5.2. Each organization sends their logs which are labelled as $r_1, r_2, ..., r_6$ to trusted party (TP), in step 1. TP anonymizes the data according to 2-diversity criteria and sends the anonymous outputs a_1, a_2, a_3 to CILMO in step 2. Assume that CILMO decided to warn the organizations about the DNS Zone Transfer attack because of its criticality. CILMO chooses the record among the anonymous records which has this attack type in the set of classification attributes. CILMO creates w_1 from a_1 by stripping off all organization attributes and all classification information except "DNS Zone Transfer" and sends w_1 to IDS management servers of organization 1 (O1) and organization 2 (O2) in steps 3 and 4. In step 5, O1 and O2 query whether an intrusion log exist about the profile given in $w1$ and deduce that whether the corresponding warning is related with their organization.

A drawback of this mechanism is that the organization O2, which decides the warning does not belong to itself in the above example, also gets the profile information of intrusion occurred for O1. O2 learns that a DNS Zone Transfer attack is performed at 11:00 by 194.100.4.4 to an organization; but it cannot learn that the targeted organization is O1.

If trusted party is allowed to store mapping information between original data

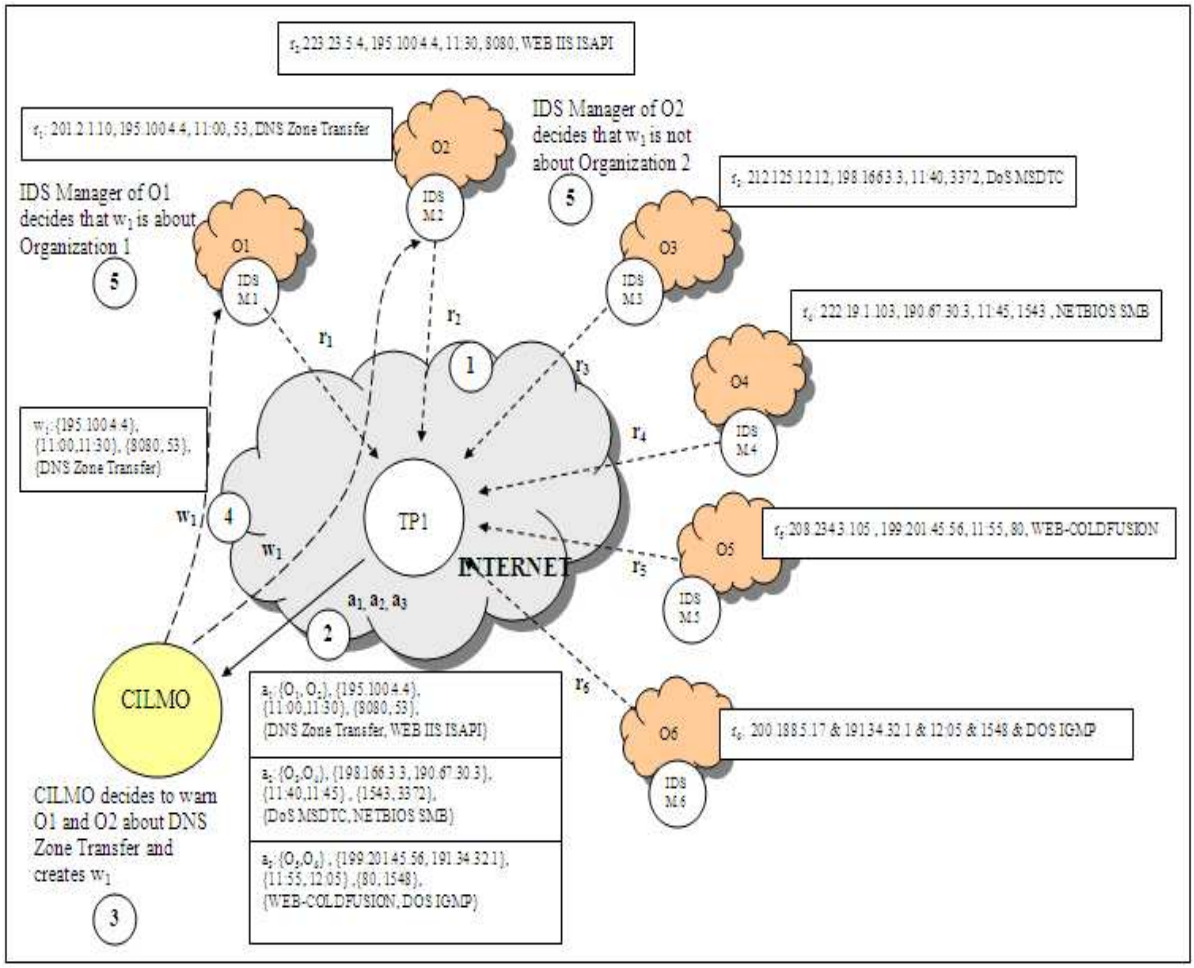


Figure 5.2: Warning Mechanism with the requirement that trusted party does not store any information

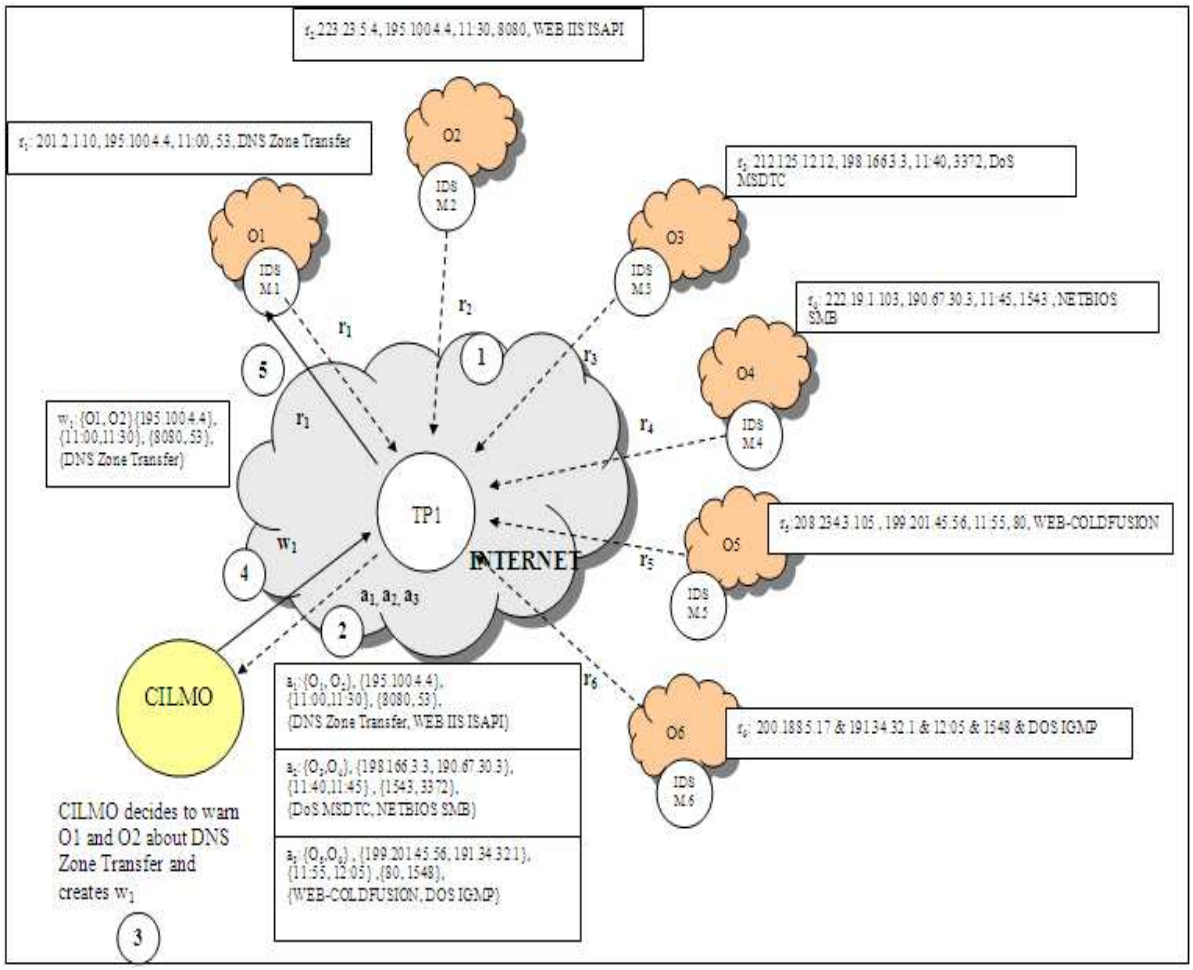


Figure 5.3: Warning Mechanism with the requirement that trusted party can store information

and anonymous output, after deciding the warning message, CILMO sends w_1 to TP. TP finds the exact matching record r_1 with the DNS Zone Transfer attack and relays r_1 to O1 as shown in Figure 5.3. In this method, an organization does not learn anything about the intrusion logs of other organizations.

5.4 Performance Evaluation of *l*-ACM

In this part, the performance of l-ACM is evaluated in terms of *information loss* and *average response time* of intrusion log records. Average response time, T_{avg} shows

the average of the amount of times between generation of log at owner organization and arriving corresponding warning to that organization from CILMO.

In our experiments, each organization generates intrusion log in a such a way that all attributes of logs are formed using uniform distribution. The log generation time for i^{th} log record is represented as t_g^i. Log generation rate, lgr, which is the number of produced logs per minute, is a predetermined parameter that adjusts the speed of log generation. It is assumed that each organization uses the same log generation rate. All log records generated in one minute is collected at organization site and they are sent to CILMO at the end of that minute. Therefore, i^{th} log record waits $60 - t_g^i$ seconds at organization site before sending to CILMO.

After CILMO receives logs, anonymization operations take place by using l-ACM. Anonymization is completed in several steps. In each step, data set which includes only one record from each organization is chosen among the received logs and they are anonymized. Otherwise, if we include more than one record from each organization, an anonymity set may contain more than one record belonging to same organization which violates l-diversity property. Restriction of one record from the same organization actually means that number of steps needed for completion of anonymization is numerically equal to log generation rate. The duration of m^{th} anonymization step is represented as t_a^m

In l-ACM, we use a record selection method for preparing input data of each anonymization step. Our method chooses an initial record from the first organization. For each other organization, logs of an organization are compared with the record of first organization and the one having most similarity is chosen as an input record in that step.

Anonymized outputs are analyzed by CILMO. If analysis results require to send a warning to appropriate organization, warnings are sent by using one of the methods given in Section 5.3. In performance calculations, parameter called log analysis time,

t_l, is used for log analysis of one log record at CILMO. Warnings are sent after this analysis time has passed.

Transmission time needed for transferring one log record from organization to CILMO and time for transferring one warning to organization is represented as t_r. In average response time calculations, we assume that for each log record, CILMO sends a warning message.

Average response time for a log record is calculated as given in Equation 5.1. We assume that total number of input record is n.

$$T_{avg} = (60 - t_g^i) + \sum_{s=1}^{s=m} t_a^m + t_l + 2.t_r. \tag{5.1}$$

The effects of changes in parameter l and lgr with respect to information loss and response time performances of l-ACM are investigated via simulations. Experiments are performed in a laptop having 1.20 GHz CPU and 2GB RAM. Intrusion data is synthetically generated. A java implementation is developed for data generation, application of l-ACM and evaluating the results.

k-ACM calculates the information loss according to Equation 3.5. In this formula, F_{ij} is the total number of bits having value '1' for the i^{th} record of j^{th} attribute. On the other side, l-ACM produces anonymized output with attribute value sets instead of bit strings. Therefore, l-ACM uses the size of attribute value set (which means the number of distinct elements in the set) instead of F_{ij}.

There are 100 distinct attackers in the network. The number of distinct values for intrusion classification is 15 and the number of slots for time value is 100. There are distinct 10 destination service in the data set. According to these parameters, maximum information loss is calculated as 5.54 via the help of Equation 3.5. If anonymized outputs contain data entries consisting of all possible data values, it has the maximum information loss. For example, including all distinct attackers in the output generates an information loss of log (100). Overall information loss is

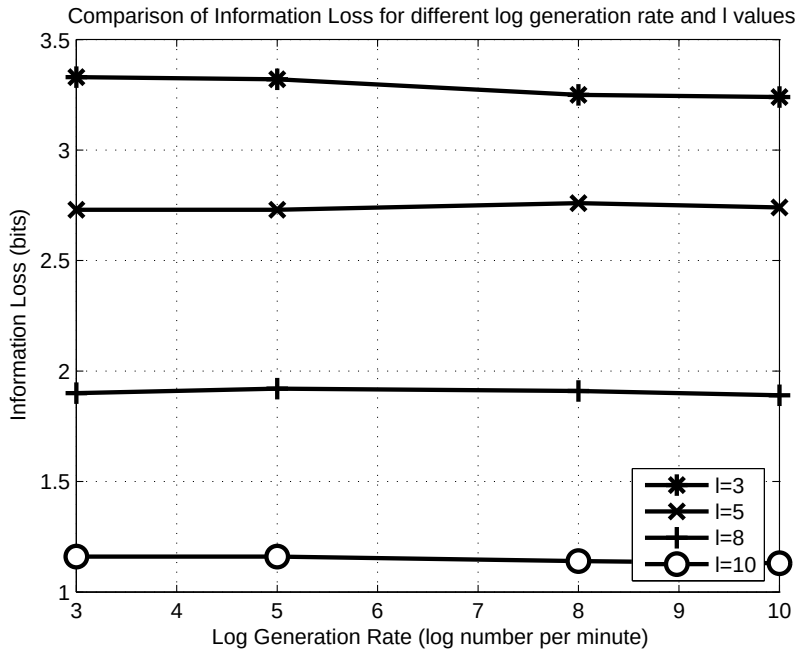


Figure 5.4: Effects of lgr and l on Information Loss

actually the average of maximum information loss caused by each attribute.

Effects of lgr and l values on information loss results is given in Figure 5.4. In these experiments, number of organizations which send their logs to CILMO is fixed to 500. As shown in Figure 5.4, increase in lgr does not effect information loss values for each l value.

Effects of lgr and l values on average response time are given in Figure 5.5. In this experiment, number of organizations is also fixed to 500. It is observed that average response time increases as lgr increases for each l values. There is a linear relationship between average response time and lgr values. Since lgr also determines the number of anonymization steps performed at CILMO, increase in the number of steps increases the time for anonymization operations. For the same lgr, we get higher average response time values for higher l values due to need for much more processing in hierarchical clusterings.

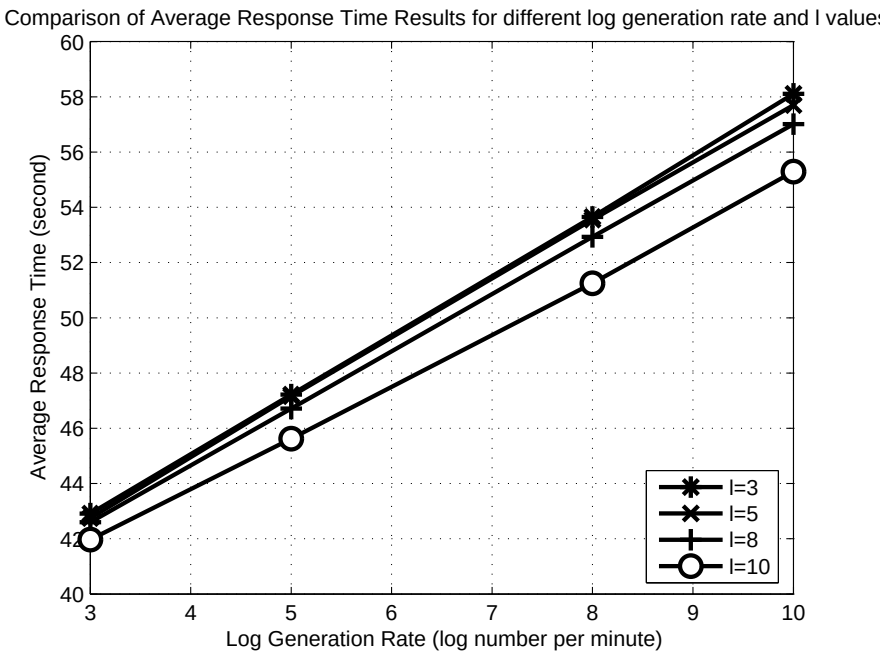


Figure 5.5: Effects of lgr and l on Average Response Time

Effects of the changes in number of organizations are analyzed. Figure 5.6 shows the effects of organization number to information loss. Figure 5.7 analyzes average response time results of l-ACM with different number of organizations. In these experiments, l and lgr values are fixed to 5 and 8 respectively.

From Figure 5.6, it is deduced that information loss value exponentially decreases as the number of organization increases. Since anonymization is performed among bigger sets of log records in higher organization numbers, l-ACM has the possibility to find more similar records during hierarchical clustering.

Figure 5.7 shows that higher number of organizations causes higher response times. There exists an exponential increase in response times. Increase in organization number means in each anonymization step, higher number records are given as an input to l-ACM. Running time of k-ACM is given as $O(n^2 logn)$ in 3.8 of Chapter 3. Here n represents the number of input records inserted to k-ACM. Since l-ACM

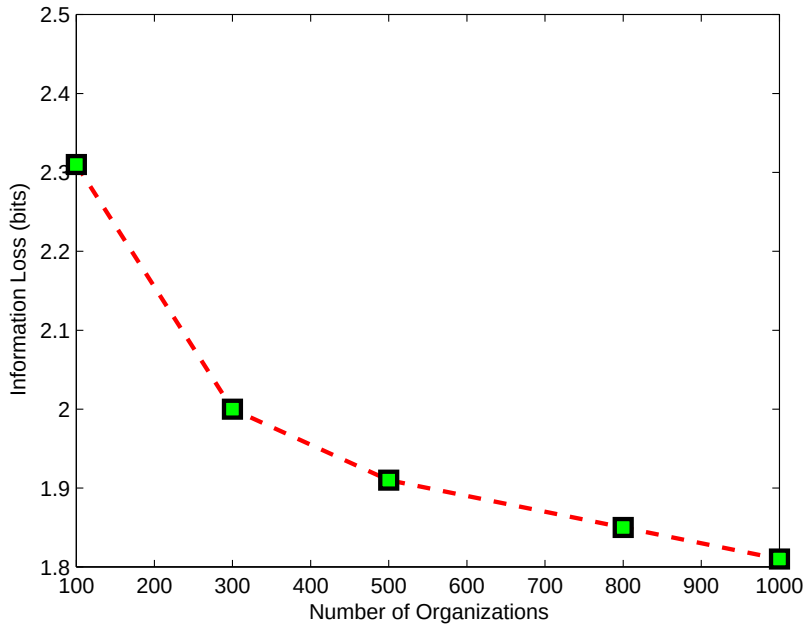


Figure 5.6: Effects of Organization Number on Information Loss

bases on k-ACM, an exponential increase in the average response time is expected as shown in Figure 5.7.

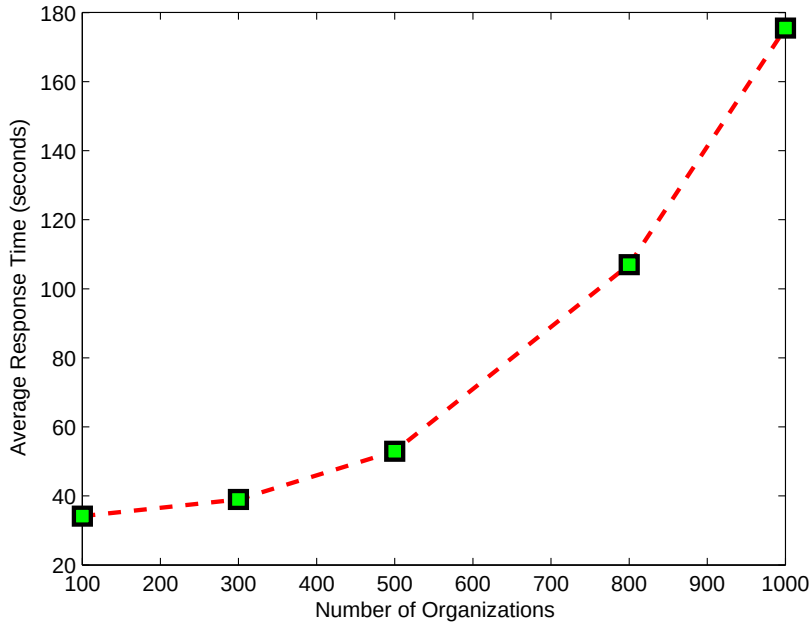


Figure 5.7: Effects of Organization Number on Average Response Time

Chapter 6

Conclusions

Network and database communities approach to privacy problem from different aspects. In one side, network community mostly thinks that hiding sender and/or receiver in network communications is the prominent privacy problem. Privacy threat models presented by this community are based on concealing communication profiles against traffic analysis attacks. These attacks focus on external threats like global or local eavesdropping. On the other side, database community uses "data" per se as the subject of privacy. They bring solutions for privacy preserved storage and sharing of data. However, privacy models are not suitable for the actual needs of network applications, where data is gathered from users and relayed to data collectors. There are some models that fully trust data collection parties which are not very realistic in most of the cases. There are some other models that consider data collector as an un-trusted entity. However, in such models, privacy preservation is provided by sending perturbed data to the data collector, which limits the types of analysis that can be performed by data collector.

In general, privacy models developed by database community do not map to user-centric network applications. Network community deals with sender-receiver anonymity, which may not be the real threat in most of data collection applications.

In this thesis, we propose a privacy preservation framework for user centric data collection applications. Our framework encompasses all stages of data collection,

including collection of data from users and sending to the data collection points. It proposes solutions for applications that may have more than one data collector points with different level of privacy requirements. Proposed framework brings solutions for possible requirements of bandwidth limitation and energy consumption.

We proposed a method, k-ACM (k-Anonymization Clustering Method) which makes data k-anonymized. k-ACM is based on UPGMA (Unweighted Pair Group Method with Arithmetic Mean), a well-known bottom-up hierarchical clustering mechanism. Conditional entropy notion of information theory is adapted to be used as the distance function, which calculates the information loss during each clustering process. The same notion is also used for calculating the anonymity level of k-anonymized data and for evaluating the results of experiments in terms of information loss. Additionally, this thesis introduces dynamic taxonomy tree idea for the generalization operation in order to decrease the data utility of collected data.

Our framework is applied for two types of data collection applications: (i) privacy preserved data collection in wireless sensor networks (WSNs), (ii) preservation of organizational privacy during collection of intrusion detection logs from different organizations.

Minimization of energy is an important criteria in WSNs. Therefore, privacy preserved data collection applications in these networks also has to deal with this issue. Proposed framework is adapted for energy minimization. Our framework is applied to two different WSN network models. First model preserves privacy against semi-trusted sink and eavesdropper. Second model consists of multiple sinks which require different levels of privacy.

First network model implies two levels of k-anonymity: 1) against the semi-trusted sink; 2) against eavesdroppers. This network model assumes the sink as a semi-trusted entity, so that data received by sink must be at least $k1$-anonymous.

To protect the data against eavesdroppers, data transmitted in 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 environment and application requirements. For example, if the possibility of eavesdropper threat is high and security of sink is provided, then the WSN designers can choose higher $k2$ and lower $k1$ values.

There is a tradeoff between data quality (in terms of information loss) received at the sink and energy consumption. Quality of data reduces with generalizations since data is irreversibly perturbed via generalizations and this causes information loss.

The mechanism that we use in this model is Proposed method, mk-ACM (Modified k-Anonymization Clustering Method) which uses the idea of k-ACM iteratively two times. It firstly makes the data $k1$-anonymous by generalization operations and continues to $k2$-anonymization by generalization and encryption operations. The output size of $k2$-anonymization stage can be adjusted by a pre-determined threshold value, *output enlargement factor*. As this ratio increases, the size of output gets larger and more encryption operations are performed. During this enlargement, mk-ACM selects the most suitable data portions for encryption in order to minimize the information loss as much as possible. Increase of 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 becomes higher since data is not perturbed widely. If the ratio decreases, quality of data becomes lower but the system consumes considerably small amount of energy. In fact, mk-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 energy saving 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 information loss and/or energy saving restrictions of the WSN dictate. For example, our analysis shows that given the sink is to receive 4-anonymous data (i.e. $k1 = 4$) and 12-anonymous data is required against eavesdroppers (i.e. $k2 = 12$), and if the network can tolerate an information loss of entropy value, 0.37, then WSN designers can pick output enlargement factor as 0.4 that causes to save 43% energy while providing the required anonymity levels.

The second network model considered in WSNs states that there exists multiple sinks and each sink has different level of privacy requirements. For this model, we propose a method called Ik-ACM (Iterative k-Anonymization Clustering Method) which is also built-on our k-ACM method. Proposed Ik-ACM reduces energy consumption while fulfilling the required different privacy levels of sinks. Ik-ACM Method uses encryption operations with generalization operations in order to have one common anonymized output. This common output enables us to multicast the same message to all sinks. Multicasting of this output enables WSN to reduce amount of energy consumed for transmitting it to the sinks. In WSN, each local region has one group head node. They gather event data from sensors of their local region, anonymize it and send it to the all sinks. According to the positions of group head nodes and their distances to sinks, multicasting can be better alternative for some of the group head nodes. Each group head node decides whether multicasting is appropriate for itself or not. If it decreases energy consumption, group head node uses it. Multicasting method degrades the quality of data gathered by some sinks. Here, there is a trade-off between data loss and energy consumption. WSN designers has to decide about the trade-off between energy saving and information loss. We analyze this trade-off for different sized WSN topologies and for different sink locations. Our analyses show that, in a WSN having two sinks, it is possible to save 32% of energy however the loss of data utility received by one of the sink

increases to 1.36 from 0.88 in some topologies.

In the scope of this thesis, we also apply the proposed framework for an application in which intrusion logs are collected from different organizations by a central intrusion log management office. This office is tasked for determination of overall security posture of whole organization ecosystem, designation of security status of monitored organizations, and give feedbacks or warnings about critical intrusions to organizations. Privacy threat model states that the collected log has to have l-diversity property. This means, any administrator of central office cannot deduce the exact classification type of intrusion log among l classification types. A modified version of k-ACM, l-ACM (l-Diversity Anonymous Clustering Method), is proposed for this purpose. Different warning mechanisms are presented according to the security requirement whether trusted parties are allowed to temporarily store network traffic.

Bibliography

[1] "Health insurance portability and accountability act." http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf.

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

[3] A. Gkoulalas-Divanis and V. S. Verykiosc, "An overview of privacy preserving data mining," *Crossroads* **15**(4), 2009.

[4] C. C. Aggarwal and P. S. Yu, *Privacy Preserving Data Mining: Models and Algorithms*, ch. A General Survey of Privacy Preserving Data Mining Models and Algorithms.

[5] D. M. Carlisle, M. L. Rodrian, and C. L. Diamond, "California inpatient data reporting manual, medical information reporting for california," tech. rep., Office of Statewide Health Planning and Development, July 2007. 5th Edition.

[6] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proceedings of 17th ACM SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems*, p. 188, 1998.

[7] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *Proceedings of 22nd International.*

Conference on Data Engineering, p. 24, ICDE, 2006.

[8] M. T. Truta and V. Bindu, "Privacy protection: p-sensitive k-anonymity property," in *Proceedings of the Workshop on Privacy Data Management*, p. 94, Workshop on Privacy Data Management, In Conjunction with 22th IEEE International Conference of Data Engineering (ICDE), (Atlanta, Georgia), 2006.

[9] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," CERIAS Tech. Report 2007-78, Purdue University, 2007.

[10] A. Meyerson and 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*, pp. 223–228, June 2004.

[11] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigraphy, D. Thomas, and A. Zhu, "k-anonymity: Algorithms and hardness," technical report, Stanford University, 2004.

[12] A. Pfitzmann and M. Khntopp, "Anonymity, unobservability, and pseudonymity- a proposal for terminology," pp. 1–9, International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, 2001.

[13] A. Pfitzmann and B. Pfitzmann, "Networks without user observability-design options," Eurocrypt, Springer-Verlag, (Berlin), 1986.

[14] J. F. Raymond, "Traffic analysis: Protocols, attacks, design issues and open problems," in *Proc. Workshop on Design Issues in Anonymity and Unobservability*, pp. 10–29, 2001.

[15] D. Chaum, "The dining cryptographers problem: Unconditional sender and

receipent untraceability," *Journal of Cryptology* **1**(1), pp. 65–75, 1998.

[16] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the Associations for Computing Machinery* **24**(2), pp. 84–88, 1981.

[17] D. Kesdogan, J. Egner, and R. Busckhes, "Stop-and-go mixes providing probabilistic security in an open system," in *Proceedings of the Second International Workshop on Information Hiding*, pp. 83–98, 1998.

[18] A. Back, U. Mller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Proceedings of the 4th International Workshop on Information Hiding*, pp. 245–257, 2001.

[19] "Anonymizer." http://www.anonymizer.com.

[20] M. G. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications* **16**(4), pp. 482–494, 1998.

[21] M. K. Reiter and R. A.D., "Anonymous web transactions with crowds," *Communications of the ACM* **42**(2), pp. 32–48, 1999.

[22] C. Shields and B. Levine, "A protocol for anonymous communication over the internet," in *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pp. 33–42, 200.

[23] J. Kong, X. Hong, and M. Gerla, "A new set of passive routing attacks in mobile ad hoc networks," **2**, pp. 796–801, Military Communications Conference, IEEE Milcom, 2003.

[24] D. B. John and D. Maltz, "Dynamic source routing in ad hoc wireless networks,"

Mobile Computing **353**, pp. 153–181, 1996.

[25] C. E. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," pp. 90–100, IEEE WMCSA '99, 1999.

[26] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Proc. of the 21st Int'l Conf. on Data Engineering*, pp. 205–216, April 2005.

[27] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowledge-Based Systems* , 2002.

[28] K. Wang, P. Yu, and S. Chakraborty, "Bottom-up generalization: A data mining solution to privacy protection," in *Proc. of the 4th IEEE International Conference on Data Mining*, pp. 249–256, November 2004.

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

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

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

[32] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal* **27**, pp. 379–423, 623–656, 1948.

[33] A. S. Voulodimos and C. Z. Patrikakis, "Quantifying privacy in terms of entropy for context aware services," *Journal of Identity in the Information Society* ,

2009.

[34] B. C. M. Fung, K. Wang, L. Wang, and P. C. K. Hung, "Privacy-preserving data publishing for cluster analysis," *Data & Knowledge Engineering* **68**(6), pp. 552–575, 2009.

[35] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," Workshop on Privacy Enhancing Technologies, 2002.

[36] I. Kojadinovic, "Agglomerative hierarchical clustering of continous variables based on mutual information," *Computational Statistics & Data Analysis* **46**, pp. 269–294, 2004.

[37] P. Andritsos and V. Tzerpos, "Software clustering based on information loss minimization," in *Proceedings of 10th Working Conference on Reverse Engineering*, p. 334, WCRE 03, 2003.

[38] A. Kraskow and P. Grassberger, "Mic: Mutual information based hierarchical clustering," *Information Theory and Statistical Learning* , pp. 101–123, 2009.

[39] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," pp. 31–42, First International Conference On Mobile Systems, Applications, Services (MobiSYS), 2003.

[40] M. Gruteser, G. Schelle, A. Jain, A. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *Proceedings 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS)*, **9**, p. 28, 2003.

[41] A. Perrig, R. Szewczyk, V. Wen, W. Culler, D. Culler, and J. D. Tygar, "Spins: Security protocols for sensor networks," in *Proceedings of The Seventh Annual International Conference On Mobile Computing and Networking*, *189-199*, 2001.

[42] B. Przydatek, D. Song, and A. Perrig, "Sia: Secure information aggregation in sensor networks," in *Proceedings of the First International Conference On Embedded Networked Sensor Systems*, 2003.

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

[44] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "On providing anonymity in wireless sensor networks," in *Proceedings of the Tenth International Conference on Parallel and Distributed Systems*, (411), ICPADS 04, 2004.

[45] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," pp. 109–117, The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005.

[46] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing* **7**(1), 2008.

[47] "Deepsight threat management system." https://tms.symantec.com/Default.aspx.

[48] "Internet storm center." http://isc.sans.org/.

[49] G. Minshall, "Tcpdriv command manual," 1996.

[50] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: Measurment-based security evaluation and a new crypthography-based scheme," IEEE International Conference on Network Protocols, 2002.

[51] A. Slagell, Y. Li, and K. Luo, "Sharing network logs for computer forensics: A new tool for the anonymization of netflow records," Computer Network Forensics Research Workshop, held in conjuction with IEEE SecureComm, 2005.

[52] J. Zhang, N. Borisov, and W. Yurcik, "Outsourcing security analysis with anonymized logs," 2nd IEEE Intl. Workshop on the Value of Security through Collab., 2006.

[53] C. D. Michener and R. R. Sokal, "A quantitative approach to a problem in classification," *Evolution* **11**, pp. 130–162, 1957.

[54] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks* **52**(12), pp. 2292–2330, 2008.

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

[56] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of 2003 Symposium on Security and Privacy*, p. 197, 2003.

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

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

[59] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable

and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, 2000.