UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Security in Mobile Networks: Communication and Localization

Autore:

Francesco Giurlanda _____

Relatore:

Prof. Gianluca Dini _____

Anno 2013
SSD ING-INF/05

Sommario

Attualmente le reti mobili sono ovunque. Il mondo è diventato sempre più dipendente da servizi mobili e wireless, ma la rapida crescita di queste tecnologie solitamente sottovaluta gli aspetti di sicurezza che ne derivano. Tanto più servizi mobilie e wireless crescono, più le debolezze nelle infrastrutture di rete diventano chiare. Uno dei problemi riguarda la riservatezza. Le tecnologie wireless possono ridurre i costi, incrementare l'efficienza e rendere informazioni importanti più accessibili. Ma, ci sono anche dei rischi. Senza dovute precauzioni, queste informazioni possono essere lette e modificate da utenti non autorizzati. Ci sono molte soluzioni, più o meno efficaci, per proteggere i dati da utenti non autorizzati. Ma, un specifica applicazione potrebbe voler distinguere più flussi dati tra utenti autorizzati. Proteggere la riservatezza di queste informazioni tra sottoinsiemi di utenti non è un problema banale.

Un altro problema è l'affidabilità del servizio wireless. Sistemi con più veicoli composti da Autonomous Guided Vehicles (AGVs) sono ampiamente usati per i trasporti industriali di sistemi logistici e manifatturieri. Questi veicoli costituiscono una rete wireless mobile per scambiare informazioni al fine di coordinare i compiti e i movimenti. La distribuzione affidabile di queste informazioni è un'operazione cruciale, perché gli AGV potrebbero acquisire una visione inconsistente del sistema che porta al fallimento del coordinamento. Ciò ha delle evidenti implicazioni di sicurezza.

Andando più in profondità, anche se il sistema di distribuzione è affidabile, le informazioni di posizionamento, trasmesse da ogni veicolo, devono essere corrette. Solitamente, i veicoli ottengono le informazioni di posizionamento attraverso una rete wireless secondaria, come il GPS. Tuttavia, il diffuso GPS civile è estremamente fragile in uno scenario ostile. Una stima non sicura di distanza o posizione potrebbe produrre problemi di sicurezza quali accessi non autorizzati, denial of service, furti, violazione dell'integrità del sistema con possibili implicazioni per la sicurezza e disastri intenzionali.

In questa tesi, affronteremo questi tre problemi, proponendo una soluzione originale per ciascuno di essi.

Abstract

Nowadays the mobile networks are everywhere. The world is becoming more dependent on wireless and mobile services, but the rapid growth of these technologies usually underestimates security aspects. As wireless and mobile services grow, weaknesses in network infrastructures become clearer. One of the problems is privacy. Wireless technologies can reduce costs, increase efficiencies, and make important information more readily and widely available. But, there are also risks. Without appropriate safeguards, these data can be read and modified by unauthorized users. There are many solutions, less and more effective, to protect the data from unauthorized users. But, a specific application could distinguish more data flows between authorized users. Protect the privacy of these information between subsets of users is not a trivial problem.

Another problem is the reliability of the wireless service. Multi-vehicle systems composed of Autonomous Guided Vehicles (AGVs) are largely used for industrial transportation in manufacturing and logistics systems. These vehicles use a mobile wireless network to exchange information in order to coordinate their tasks and movements. The reliable dissemination of these information is a crucial operation, because the AGVs may achieve an inconsistent view of the system leading to the failure of the coordination task. This has clear safety implications.

Going more in deep, even if the communication are confidential and reliable, anyway the positioning information could be corrupted. Usually, vehicles get the positioning information through a secondary wireless network system such as GPS. Nevertheless, the widespread civil GPS is extremely fragile in adversarial scenarios. An insecure distance or position estimation could produce security problems such as unauthorized accesses, denial of service, thefts, integrity disruption with possible safety implications and intentional disasters.

In this dissertation, we face these three problems, proposing an original solution for each one.

Contents

List of Figures

List of Tables

1

Introduction

Networks of mobile devices have grown rapidly over the past decade, reaching maximum expression with the well known cellular network. Nevertheless, there are many other types of mobile networks. Networks of Automated Guided Vehicles (AGVs) largely used for industrial transportation in manufacturing and logistics systems. Mobile ad-hoc networks (MANET) are used to provide crisis management services applications, for example in a disaster recovery where the entire communication infrastructure is destroyed and resorting communication quickly is crucial. The MANET have also military applications, i.e. some of the essentials requirements of a combat operations include network deployability, network security and high mobile connectivity. Personal Area Networks (PAN) are designed to support a set of communicating devices within range of few meters, eliminating the need of wires.

All these technologies have in common the type of communicating system, that is wireless. Even if the implementations of the wireless communication system can be different, all of them subject the system to many security issues. These issues depend on the very nature of a wireless system. In a wired system, only the devices which are plugged in the network are able to access the network communication, but in a wireless system every device with an appropriate receiver can access to the network communication.

Simultaneously with the development of wireless communication systems, the researchers have developed solutions, more and less effective, to resolve these security issues. For instance, the Wireless Equivalent Privacy (WEP) is the first security choice presented (1999) and it aims at achieving the same security level of a wired network. It is still widely in use, but WEP has been demonstrated to have numerous flaws and has been deprecated in favor of newer standards, such as Wi-Fi Protected Access 2 (WPA2). These solutions solve the control access issues, but they cannot protect private communication between valid users, i.e. when a user is authorized to

access a wireless network, he can read also the transmission that are not addressed to him. Moreover, an unauthorized user can exploit other weaknesses of the wireless communication systems which do not require the access to the network such as jamming-and-replay attacks, wormhole attacks and packets injection. The shown weaknesses are the consequence of the broadcast nature of the wireless medium.

1.1 Identify the Problems

In this work we deal with three problems that are direct consequence of the broadcast nature of the wireless medium: privacy, reliability and localization. These problems afflict every type of mobile network. It is easy understand the implications between the nature of the wireless medium and the problems of privacy and reliability. Otherwise, we tend to underestimate the security problems related to the localization system. During the designing of a mobile network system, the localization task is assigned to a well-known technology such as GPS. It is considered as a black box that provide the position of the device. But, also the localization system uses a wireless network to estimate the position, thus it is affected by the same problems of wireless networks. Below, we explain more in detail the implication of these three security problems.

1.1.1 Privacy

Let us suppose that a large group of mobile devices communicate together. The group communications must be confidential. The simplest solution leverages on cryptography, i.e. each group member keeps a secret group key which is used to encrypt/decrypt the group communication. Efficiently managing cryptographic keys for large, dynamically changing group is a difficult problem. Group secrecy requires that a device must be able to encrypt and decrypt communication within a given group only while the device is member of that group. It follows that upon a device *joining* the group, a new group key must be distributed to all group members so that the new member cannot decrypt previous group communication (*backward security*). Furthermore, upon a device *leaving* the group, the current group key must be revoked and a new one must be distributed to the remaining group members so that the leaving device cannot read future group communication (*forward security*). It is obvious that it is impossible stop the system to update the group key at each membership change, because it makes the service discontinuous. These requirements can be fulfilled by *rekeying algorithms*. Intuitively, a keying algorithm distributes the group key to all group members which use it to encrypt and decrypt broadcast messages. When

a new member joins or a current member leaves the group, the current group key is revoked and a new one distributed in a efficient way and without stopping the system.

The problem of group key management becomes more complex when the whole group can be subgrouped according service specifics. For instance, different classes of devices could be subgrouped according their features (aerial vehicles, terrestrial vehicles, control devices...). Therefore, the rekeying algorithm must be able to manage the confidentiality at sub-group level in a very general *multi-group* model with many subgroups, hierarchically organized, and possibly overlapping. In this model, backward and forward security must be guaranteed at sub-group level.

1.1.2 Reliability

We consider a common type of mobile network largely used for industrial transportation in manufacturing and logistics systems: a Multi-vehicle system composed of Autonomous Guided Vehicles (AGVs). This system offers potential advantages in terms of task speedup, robustness and scalability. For instance, a typical function of a multi-AGVs system consists in transporting raw or semi-finished material from warehouse to production lines. However, deployment of a team of AGVs raises management and coordination problems such as collision avoidance, conflict resolution, and shared resources negotiation. The vehicles need exchange environment information to achieve these tasks. It has clear safety implications. If dissemination is not secure and reliable, an adversary may inject fake messages or simply delay a message so leading AGVs to achieve wrong and/or inconsistent views. Once again this may cause the coordination task to fail. Moreover, the system is subjected to many risks also in an environment without adversaries. When an AGV broadcasts its state, an accurate and timely notion of its neighborhood is crucial to avoid collisions, i.e. the AGV have to track which neighbors have received such state and which have not and thus need a re-transmission.

The AGVs exchange state information using a communication protocol that operates over an IEEE 802.11 wireless network technology [24]. This technology is rapidly expanding in industrial scenarios due to its recent improvements in terms of hardware costs, transmission speed, and simplicity and flexibility of deployment [13] . However, it lacks any reliable broadcast service. If two AGVs transmit a state packet at the same time, the packets collide and no-one will receive the message. We need same solution that improves the reliability of the broadcast service over an IEEE 802.11 wireless network.

1.1.3 Localization

The measurement of the distance between two electronic devices, and then the position, is crucial for many practical applications. Many techniques have been proposed over the years [34]. All these techniques fail in the presence of an adversary that wants to disrupt the distance measurement process. Even the well known and widespread civilian Global Positioning System (GPS) is extremely fragile in adversarial scenarios [30]. Secure location estimation has a plethora of applications including coordination of AGVs [20, 18] and geographical routing in a mobile network [29, 64]. For all these applications, an insecure distance or position estimation could produce security problems such as unauthorized accesses, denial of service, thefts, integrity disruption with possible safety implications and intentional disasters.

Desmedt [14] first introduced the problem of secure location verification and showed that it cannot be solved by solely using cryptography. Brands and Chaum [6] proposed the first *secure distance-bounding* protocol. Since then, many variants have been proposed in the literature [7, 41, 53]. These protocols leverage on both the unforgeability of authenticated messages and the upper bound of the communication speed that is the speed of light. They prevent *distance reduction*, i.e., an adversary cannot make a device appear closer than it really is. The resistance against distance reduction is an important requirement for all the application scenarios involving secure proximity verification [23, 19, 22, 26]. A common example is the problem of proximity-based access control. Let us suppose an RFID card performing an authentication protocol with a reader. If the card correctly performs the protocol, the reader will open a door of a building. An adversary can trick the system by establishing a relay link between the reader and a far away legitimate card, owned by an unaware user. The card correctly performs the authentication protocol via the relay link, and the reader opens the entrance. This attack is known as *mafia fraud*. Along with the correctness of the authentication, the reader has to check even that the card is within a security distance. However, if such a distance measurement is made with insecure methods, the adversary can still break the system. In particular she can perform a distance reduction attack to deceive the reader into believing that the far away card is in the proximity.

The relevance of the secure proximity verification eclipsed the dual problem: the *distance enlargement* attack. By this attack, an adversary makes a device appear farther than it really is. The resistance against both reduction and enlargement attacks is important whenever we want to securely estimate a distance, rather than a proximity. Let us suppose a distributed system that monitors the movement of autonomous guided vehicles. The system relies on distance information to avoid collisions between

vehicles. An example of such systems is in [18]. If an adversary is able to make a distance appear larger than it really is, the system could not take collision-avoidance countermeasures in time. This could cause collisions between vehicles, and consequent loss of money and safety threats. Secure distance estimations are extremely useful in trilateration techniques too. These techniques use the distances measurements from at least three anchor nodes, whose positions are known, to estimate the position of a fourth node. If an adversary can enlarge one or more distance measurements, she is able to disrupt the whole positioning process.

1.2 Contributions of This Dissertation

In this dissertation we face the problems that afflict the communication (privacy and reliability) and the positioning (secure localization) in a mobile network. we propose a centralized *multicast key distribution* (MKD) scheme, named Multi-Group Logical Key Hierarchy (MG-LKH), that addresses the key management in a general multi-group model. MG-LKH is scalable in storage, computing, and, especially, communication with respect to the number of users and the number and size of sub-groups. The scheme exploits a logical hierarchy of keys, and outperforms traditional rekeying schemes when they are adapted to the multi-group scenario. Moreover, we are going to show that our scheme has better performance of existing solutions to handle key management issues associated with multi-group model.

At a later stage, we are interested in increasing the reliability of the state dissemination aimed at AGVs coordination service without influencing the broadcast traffic of other protocols. Therefore, starting from the periodic nature of the state dissemination traffic pattern, we have designed an accurate, efficient, and scalable protocol that is suitable for real-time coordination protocols. This "Neighborhood Monitoring" protocol (NMP) is *accurate* because the difference between the actual neighborhood of an AGV and the view of that neighborhood the protocol provides the AGV is negligible. Moreover NMP makes it possible to estimate a *maximum state dissemination delay* that is fundamental in real-time applications. Furthermore, NMP is *efficient* as it produces a negligible rate of collisions and consequent packet loss so solving the insidious problem that afflicts the periodic and uncoordinated data dissemination protocol, namely the overlapped transmissions. Differently from other approaches, NMP is *scalable* because state dissemination influences only actual neighbors and the dissemination rate is automatically reduced when neighbors are absent.

About the localization problem, we propose SecDEv (SECure Distance EValuation), a distance-bounding protocol able to resist to enlargement attacks based on jam-and-replay tactics [31, 60, 59]. SecDEv exploits the characteristics of wireless

signals to establish a *security horizon* within which a distance can be correctly evaluated (besides measurement errors) and any adversarial attempt to play a jam-and-replay attack is detected. We also are going to show how SecDEv improves the scalability of secure positioning techniques in terms of number of anchor nodes.

1.3 Dissertation Organization

This dissertation is structured into three chapters where we deal with the three security problems above mentioned.

In chapter 2 we present a brief state of the art about key management schemes. Subsequently, we describe MG-LKH scheme and we analyze its performance in terms of communication, storage and computation costs. We show also the results of out simulations that compare MG-LKH with the performance of the best schemes in the state of the art.

Chapter 3 explores the reliability problem, we present the problem and some solutions developed to mitigate it. Then, we introduce our protocol, named "Neighborhood Monitoring" protocol, and we consider a case-study where a group of UAVs (Unmanned Aerial Vehicles) communicate together to avoid collisions. We have simulated the case-study and we show the results in therms of reliability and scalability of the protocol.

In chapter 4 we deal with the problem of secure localization. We show the state of the art and we analyze the most known solutions. Then, we describe the our solution: a communication protocol to estimate the distance between two devices in a secure way, called SecDEv (Secure Distance EValuation). We analyzed the efficiency of our solution in terms of covered area and we compared it with an existing one, *Verifiable Multilateration* [60], which is the state-of-the-art technique for secure positioning in wireless networks.

Finally, chapter 5 contains our conclusions about the mobile networks and the three analyzed issues.

2

Privacy

Nowadays, a new class of applications based on broadcast/multicast communication is emerging. These applications include mobile networks, secure audio and video broadcasting, pay-TV, secure conferencing, wireless sensor networks. All these applications feature very large groups, e.g., thousands of users, with very dynamic membership. In these applications, users can be further sub-grouped on service basis. For instance, in a mobile network the devices can be sub-grouped according their features: control devices, sensors, aerial or terrestrial vehicles. These sub-groups may overlap. Therefore, this new class of group-oriented applications is characterised by a very general *multi-group* model with many groups, hierarchically organised, and possibly overlapping.

According these new applications, the interest in secure group communication has grown. Efficiently managing cryptographic keys for large, dynamically changing groups is a difficult problem. Group secrecy requires that a user must be able to encrypt and decrypt communication within a given group only while the user belongs to that group. It follows that upon a user *joining* the group, a new group key must be distributed to all group members so that the new member cannot decrypt previous group communication (*backward security*). Furthermore, upon a user *leaving* the group, the current group key must be revoked and a new one must be distributed to the remaining group members so that the leaving user cannot read future group communication (*forward security*) [61, 38]. Moreover, backward and forward security must be guaranteed at the level of sub-group. These requirements can be fulfilled by *rekeying*. Intuitively, a group/sub-group key is distributed to all group/sub-group members which use it to encrypt and decrypt broadcast messages. When a new member joins or a current member leaves the group/sub-group, the current group/sub-group key is revoked and a new one distributed.

The system who assures the backward/forward secrecy and manages a common decryption key to a dynamic group of authorized members over a broadcast channel is called the *group key management* (GKM). The GKM scheme provides an automatic mechanism for keys renewal. In our case, the GKM scheme is symmetric-keys based because it is the natural choice for ensuring secure access and secure data dissemination. Furthermore, the GKM is going to manage more keys, at least one for each sub-group. Taking into account that a broadcast service works with many sub-groups and a large number of users, the GKM must be *efficient* in terms of communication cost during the rekeying operation and storage and computation costs for each members and the server. At the same time, the GKM must provide a high level of *flexibility* because a user should be able to join/leave to any sub-group at any time.

According to the key status of members, two category of GKM schemes in the literature may be applied in broadcast services: the *stateless* GKM schemes [5, 21, 35, 39], called also *broadcast encryption* (BE), and the *stateful* GKM schemes [9, 8, 33, 44, 51, 62], called also *multicast key distribution* (MKD) schemes. A member in a BE scheme does not update his keys after the joining operation. In contrast, a member in a MKD scheme updates his keys by the rekeying messages broadcasts from the *group manager* (GM). The MKD schemes are more efficient and scalable than the BE schemes in context of groups with a large number of members. Both classes of GKM schemes can be divided into three different categories [47]: the *centralized* schemes with a central group controller (GC) that are in the opposite direction of the *distributed* schemes without GC. In the middle of them, the *decentralized* schemes can be regarded as the mix of previous two.

Since the researchers proposed the problem of group key management, many GKM schemes have been proposed. Relevant examples include *Simple Key Distribution Center* (SKDC) [27], *Group Diffie-Hellman* (GDH) [54], *Logical Key Hierarchy* (LKH) based algorithms [61, 62, 51, 8, 16]. Especially the approaches based on LKH facilitate rekeying operations because they turn the communication and computation costs from linear into logarithmic in the group size. These LKH schemes were designed to handle key management issues associated with single-group model. Although access control in multi-group model can be managed separately for each sub-group, using the existing GKM schemes, this leads to inefficient use of keys and does not scale well when the number of sub-groups increases. More efficient, but at the same time more complex solutions for multi-group models were proposed in [56, 25, 11]. From this point onwards, we will use *cluster* in place of sub-group.

In this chapter, we are going to present a novel centralized MKD scheme, named Multi-Group Logical Key Hierarchy (MG-LKH), that addresses the key management in a general multi-group model. MG-LKH is scalable in storage, computing, and, espe-

cially, communication with respect to the number of users and the number and size of clusters. The scheme exploits a logical hierarchy of keys, and outperforms traditional rekeying schemes when they are adapted to the multi-group scenario. Moreover, we are going to show that our scheme has better performance of the solutions already proposed to handle key management issues associated with multi-group model.

2.1 Related Works

The GKM problem has been studied in deep. In the *rekeying* procedure, GM delivers a new group key to each group member so that a leaving or joining user cannot access future or prior messages of that group. These two features are referred to as forward or backward secrecy [61, 38]. One of the first solution was Group Key Management Protocol (GKMP) proposed in [27]. It is a direct extension from unicast to multicast communication. Each group member has to share only two keys: the group key to crypt/decrypt the data traffic and a private key used for updating the group key. It is a good solution in terms of storage cost for each member but its communication cost for each rekeying procedure grows proportional to the number of members. The autors in [61, 62] were aimed at solving the communication overhead problem. They proposed a new data structure called the *logical key hierarchy* (LKH). The improvement consists in the capability of communicating with subsets of whole group during the rekeying operations. This feature facilitates group rekeying because it turns the communication cost from linear into logarthminc in the group size. LKH inspired many next works such as One-way Function Tree (OFT) [51], One-way Function Chain (OFC) [8], Secure and Scalable Rekeying Protocol (S2RP) [16]. These schemes use a key tree structure. The root node of the tree is the group key and each leaf node represents a member with its private key. The interior nodes are keys which are associated with logical security domains and are used for updating the group key (more details in Section 2.1.1). To reduce the rekeying overhead of high frequency of joining/leaving operations, [33] proposed the concept of batch rekeying. GM performs the rekeying operations periodically. Between two rekeying operations, GM collects the requests of member joining/leaving in a batch and then it performs them all together. The efficiency of tree-based schemes critically depends on whether the key tree remains balanced over time as members join or depart. In [40], the autorns faced this problem with a GKM schemes that maintains a balanced key tree during the rekeying operations.

All the above solutions need a GM that coordinates the rekeying operations, at the same time GM is a single point of failure for the system. The distributed GKM scheme is characterized by having no GM. The group key can be generated in a contributory

fashion where all members, which belong to a cluster, contribute to computetion of the group key [3, 43]. Many contributory schemes are inspired by the Diffie-Hellman (DH) key exchange protocol [15]. Group Diffie-Hellman (GDH) [54], is an extension of Diffie-Hellman key agreement protocol that supports group operations. The solution is fault tolerant, because the fault of a member does not stop the system, but the solution requires that each member knows the group membership list and, in most contributory protocol, processing time and communication cost increase linearly in term of the number of members. These two constraints make the distributed GKM schemes not suitable for large group. Because of their scalability, we avoided a distributed aprouch and built our GKM scheme on a tree-based scheme.

More recently, new GKM schemes took into account the multi-group model. Until now, a simple solution was to use one of the above described GKM scheme to maintain an independent group key for each cluster. This solution is not efficient because the performance decreases linearly with the number of involved clusters. If a member belongs to more than one cluster, he has to store a set of keys for each cluster. If the member leaves the service, hence all the clusters to which he belongs to, the number of rekeing messages is proportional to the number of involved clusters. Sun and Liu [56] developed a multi-GKM scheme with an integrated key graph that maintains keying material for all member with different supercluster. The supercluster is the set of sub-groups which the member belongs to. In [25], the autors presented Key Tree Reuse (KTR), it relies on a shared key structure and the reuse of old keys in the shared key structure, without compromising security. These two aspects reduce drastically the number of rekying messages, but KTR introduces a complex management of the histrory of the keys. In this work, we chose KTR as comparison for our results because it has good performance in term of number of reekeying messages and number of keys stored for each member. Moreover, KTR defines a model for the management of clusters and superclusters that is similar to our model.

2.1.1 Logical Key Hierarchy

LKH [61, 62] is a well-known logical key hierarchy approach that has a logarithmic communication overhead in the group size. LKH uses a hierarchical system of auxiliary keys to facilitate distribution of group key. The key tree is structured as follow: the root of the tree is the *data encryption key* (DEK) that is the group key. Each leaf node in the tree is associated to a group member and it represents the *privite key* (IDK) shared only between the group member and GM. Other keys in the tree, called *key encryption key* (KEKs), are used to encrypt and update new DEKs and KEKs. It is worth to note that DEK and KEKs are logical nodes and no member is associated at

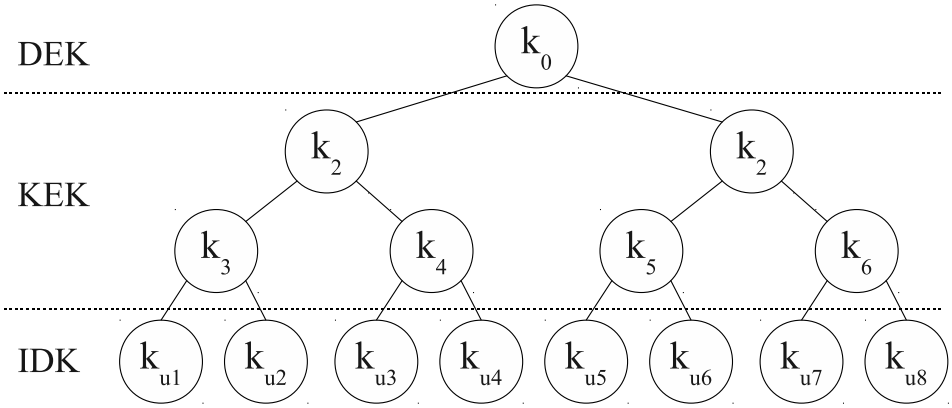

Figure 2.1. Logical Key Hierarchy

them. Each group member knows its IDK and all the keys in the path from its leaf to the root of the tree. It means that a KEK is shared with a subset of members of the whole group, moreover if it is closer to the root then the subset of members is larger. The DEK is a special KEK that encompasses all the group members.

Before starting the group key management, GM is in a phase, named *Group Initialization*, in which each member of the group holds only the IDK. In order to issue the KEKs and the DEK, GM starts from the bottom of the tree and for each KEK, GM encrypts it with KEK's children keys. Then GM broadcasts these information and only the group members that know the right keys are able to decrypt the message. The group initialization is a expensive operation in terms of number of messages because it requires $2n - 1$, but it is performed only one time at the system start.

After the initialization, GM has to be able to manage member joining/leaving operations assuring backward/forward secrecy. With reference to the Figure 2.1, let us suppose that the member $u8$ leaves the group, hence the DEK must be renewed, but all the KEKs known by $u8$ must be renewed too. GM needs to change k_0, k_2, k_6, so it is going to broadcast the following rekeying messages:

$$\left\{\overline{k_6}\right\}_{k_{u7}}, \left\{\overline{k_2}\right\}_{k_5}, \left\{\overline{k_2}\right\}_{\overline{k_6}}, \left\{\overline{k_0}\right\}_{k_1}, \left\{\overline{k_0}\right\}_{\overline{k_2}}$$

where $\overline{k_i}$ is the new key of k_i and $\{k_i\}_{k_j}$ means that k_i is encrypted by k_j. $u7$ is the only member who can decrypt the first message, so he gets $\overline{k_6}$. Second message involves $u6$ and $u5$ which obtain $\overline{k_2}$ and with third message also $u7$ receives $\overline{k_2}$. Likewise, last two messages issue $\overline{k_0}$ to the whole group exept $u8$. To explain the joining operation, let us suppose that the member $u8$ comes back into the group. GM generates the leaf node associated to $u8$ and it adds the node as child of k_6. Either in this case, k_0, k_2, k_6 must be renewed. GM is going to broadcast the following rekeying

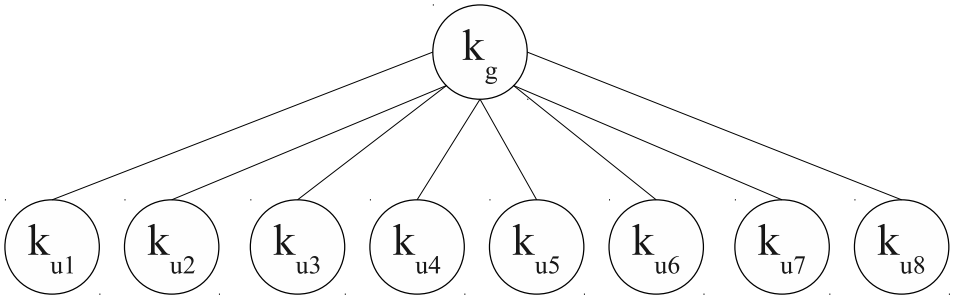

Figure 2.2. Key Star

messages:

$$\left\{\overline{k_6}\right\}_{k_{u8}}, \left\{\overline{k_6}\right\}_{k_{u7}}, \left\{\overline{k_2}\right\}_{k_5}, \left\{\overline{k_2}\right\}_{\overline{k_6}}, \left\{\overline{k_0}\right\}_{k_1}, \left\{\overline{k_0}\right\}_{\overline{k_2}}$$

The joining procedure is similar to the leaving procedure. In this case, we have one more message. First message allows the incoming member $u8$ to partecipate the rekeying procedure and to obtain the new group key $\overline{k_0}$ at the end.

LKH is a secure key management scheme and it is also efficient in terms of communication and storage costs. Each member only needs to hold $\mathcal{O}(log_2(n))$ keys and GM broadcasts $\mathcal{O}(2log_2(n))$ messages, where n is the number of members of the group.

2.1.2 Key Star

Key Star (KS) is a simple GKM scheme that we describe here because, like LKH, it takes up a significant part of our work.

KS uses few keys, GM holds the group key and a private key for each group member. With reference to Figure 2.2, the center of the star graph is a logical node associated with the group key and every radiating node is associated with the private key of the member belonging to the group. Every member keeps only two keys, the group key and his private key.

When a new member joins the group, GM generates a new group key. Then, GM broadcasts the new group key encrypted with the current group key. Furthermore, GM securely unicasts the incoming member the new group key encrypted by that member's private key. Let us suppose that a new member $u9$ joins the group in Figure 2.2, GM is going to send the following rekeing messages:

$$\left\{\overline{k_g}\right\}_{k_g}, \left\{\overline{k_g}\right\}_{k_{u9}}$$

The leaving procedure is quite different. Let us suppose that $u8$ leaves the group. GM has to renew the group key k_g, but he can deliver the new group key only using the

private keys of the remaining members. GM is going to send the following rekeing messages:

$$\left\{\overline{k_g}\right\}_{k_i}, \forall i \in G$$

Even if each member holds $\mathcal{O}(1)$ keys and the joining procedure has the same complexity $\mathcal{O}(1)$ in term of communication cost, the leaving procedure has a communication cost that is linear in the group size.

In the rest of the dissertation, we will show how we take the advantages of KS and reduce the cost of the leaving operation exploiting LKH scheme.

2.2 System Model

We consider a *broadcast service group* $U = \{u_1, u_2, ..., u_n\}$ composed of n users u_i. In a broadcast services, the data flow can be separated into different tematic flows with common features, for instance control, sensing, data into separated flows. We call *cluster* each tematic flow. Let $P = \{p_1, p_2, ..., p_M\}$ denote the set of all clusters and M is the total number of clusters. The *cluster group* ($G(p_i)$) is defined as all users which have access to the cluster.

$$G(p_i) \equiv \{u_j : u_j \in p_i\}$$

Different cluster groups can be overlapped because a user can subscribe more than one tematic flow. A set of clusters is called *supercluster* . Let $S = \{s_1, s_2, ..., s_I\}$ denote the set of all superclusters and it is easy to prove that $I \leq 2^M - 1$. The *supercluster group* ($G(s_i)$) is defined as all users who are registered to the supercluster.

$$G(s_i) \equiv \left\{ \bigcap_{p_j \in s_i} G(p_j) \right\} - \left\{ \bigcup_{p_z \notin s_i} G(p_z) \right\}$$

Distinct supercluster group cannot be overlapped because a user can belong to only one supercluster.

$$G(s_i) \cap G(s_j) \equiv \emptyset$$

The users can subscribe or change supercluster at every time. They communicate with the GM through a dedicated channel and specify the clusters that they are interested to join. Figure 2.3 shows a graph that describes the relationship between cluster, supercluster and user.

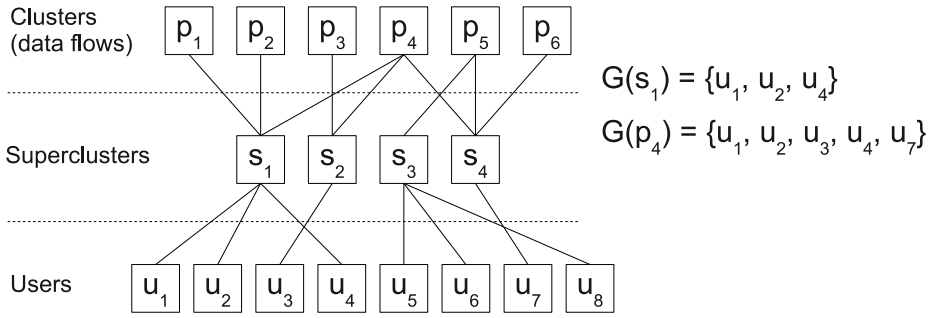

Figure 2.3. System model

2.3 Security requirements

In the applications with multiple groups that identify different multicast sessions, the operations that a user can perform are not only joining or leaving the service. Differently from the GKM schemes for single-group model, a user belongs to a supercluster and it can switch between the superclusters by adding or dropping clusters. Thus, the security requirements are more complicated than the single-group GKM schemes. From the user point of view, he can perform the following operations:

- *Service joining* is a rare operation that is associated with the initialization of an access device such as smart card. After this operation, the member cannot be able to access to any cluster yet.
- *Supercluster joining* is the operation that allows a user to access the future content of a set of clusters.
- *Supercluster leaving* is the operation that deny a user to access the future content of a set of clusters.
- *Supercluster switching* is a more complicate operation that allows a user to switch from a supercluster s_i to a supercluster s_j. The two superclusters may have one or more clusters in common. Thus, the GM has to identify tree sets of clusters into $s_i \cup s_j$. A set of clusters that the user leaves, a set of clusters that the user joins and a set of clusters that are in both superclusters and do not change.
- *Service leaving* is a rare operation that is associated with the reset of an access device such as smart card. After this operation, the member cannot be able to access the service in future. This operation is rare because usually the access devices are withdrawn from the service provider and are reassigned to a new user. It may occur that the GM has to perform this operation because the user loses the access device or the user violates the contract conditions. In these cases, the GM

has to perform also the supercluster leaving operation before the service leaving because the user has not explicitly required a supercluster leaving.

2.4 Data Encyption

As already mentioned in [56], we can use the cryptography in two different ways to achieve the above requirements. In the first method, the GM assigns a key for each cluster in P. The key K_{p_i} is shared among the users in $G(p_i)$. In this case, if a user want to subscribe to a set of clusters, then he has to require the corresponding keys from the GM. On the other hand, the GM has to securely update and distribute the clusters keys according to the above operations and assuring the backward/forward secrecy. In the second method, the GM assigns a key for each supercluster in S. The key K_{s_i} is shared among the users in $G(s_i)$. In this case, the user holds only one key K_{s_i} and it is updated only when the user changes his supercluster. Even if this method seems more efficient because each user uses only one key to dencrypt all own clusters, it introduces a relevant overhead in the service. If a cluster belongs to two or more superclusters, its data flow has to be encrypted and retransmitted as many time as the number of superclusters to which the cluster belongs to. More in detail, let us M the number of possible clusters, all the possible superclusters are $I \leq 2^M - 1$, it means that the GM may manage a huge amount of keys. Already with few clusters, for instance 20, the number of possible keys is about 2^{20}. Thus, let n denote the members of whole service, usually $I \gg n$, but at most n superclusters are valid, since the number of valid superclusters cannot be more than the number of user (one for each user). In the worst case, the GM has to manage n supercluster keys. It means that a data flow may be encrypted and retransmitted n times, with an intolerable communication overhead.

In this work, we use the first method of keys assignment because of its low number of keys managed by GM and its low data communication overhead.

2.5 Multi-Group Logical Key Hierarchy

We propose a centralized multi-group GKM scheme in which all rekeying operations are coordinated by GM. We assume that GM shares a secret key with avery user in the broadcast service group U. We denote by K_{u_i} the key that GKM shares with user u_i, and call it the *user key* (UK). We assume that this key is initially deployed by off-line means. In order to guarantee backward and forward secrecy in both broadcast service group and clusters, GM uses a logical graph of keys called Multi-Group Logical Key Hierarchy (MG-LKH). MG-LKH is the result of juxtaposition of two key graphs,

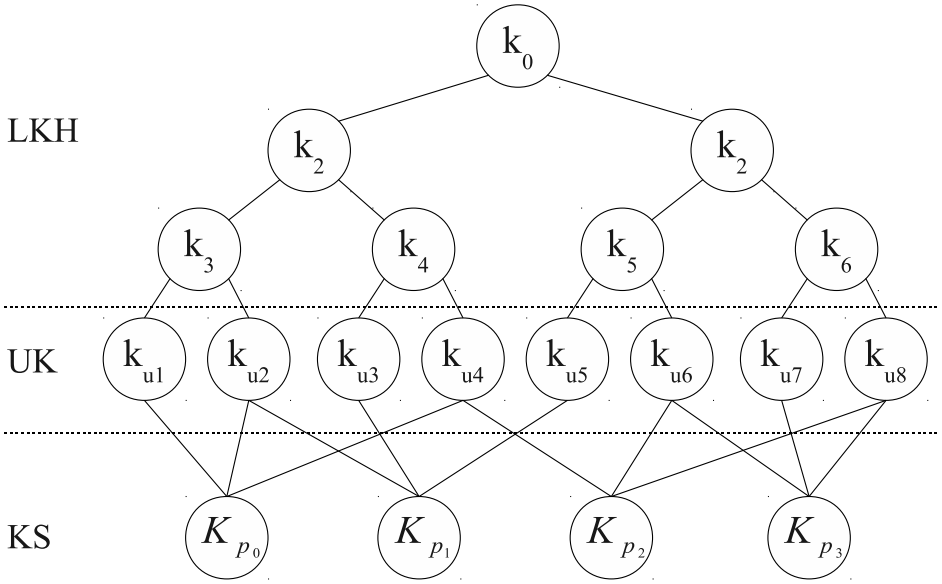

Figure 2.4. Multi-Group Logical Key Hierarchy

a Logical Key Hierarchy encompasses the whole broadcast service group, and a Key Star for each cluster. MG-LKH allows to overtake the communication overhead problem of KS (Section 2.1.2). The resulting data structure, shown in Figure 2.4, is used as follows.

Each user holds his UK, the KEKs in the path to the root of LKH tree and all the keys of the clusters gelonging to his supercluster. When a user leaves the cluster, we use the LKH tree structure to efficiently distribute the new cluster key to the cluster grup. Actually, GM broadcasts the new cluster key double encrypted by the old cluster key and a proper set of KEKs. Encrypting by the old cluster key makes it possible to exclude the users that are not cluster's members. By encrypting with the proper set of KEKs, it is possible to communicate just with the cluster group members and to exclude the leaving member as well.

For example, with reference to Figure 2.4, if u_3 want to switch supercluster from $s_1 = \{p_0, p_2\}$ to $s_2 = \{p_0\}$, it means that u_3 leaves p_2 but he continues to be member of p_0. The GM translates from a supercluster switch operation into cluster joining/leaving operations. In this case, the GM must renew K_{p_2} in u_5 and u_7. First, the GM encrypts $\overline{K_{p_2}}$ with K_{p_2} to exclude all p_2 not members. In addition, to exclude also u_3, the GM encrypts again with the KEK k_2. The resulting rekeying message is:

$$GM \rightarrow (all) : \left\{ \left\{ \overline{K_{p_2}} \right\}_{K_{p_2}} \right\}_{k_2}$$

Similarly, according to Figure 2.4 again, if u_1 switches subscription from $s_3 = \{p_0, p_1\}$ to $s_4 = \{p_0\}$, it means that u_1 leaves p_1. The GM must renew K_{p_1} in u_2 and u_4. More precisely, the GM must send two rekeying messages to notify u_2 and u_4 of the new cluster key: the first message carries $\overline{K_{p_1}}$ encrypted with K_{p_1} and KEK k_4, while the latter carries $\overline{K_{p_1}}$ encrypted with K_{p_1} and KEK k_2.

$$GM \rightarrow (all) : \left\{ \left\{ \overline{K_{p_1}} \right\}_{K_{p_1}} \right\}_{k_4}, \left\{ \left\{ \overline{K_{p_1}} \right\}_{K_{p_1}} \right\}_{k_2}$$

In case of service joining/leaving, the GM updates the tree structure adding/removing a leaf node and renewing the keys along the path from leaf node to the root node. These operations are LKH alike. The GM broadcasts only $2\log(n)$ encrypted keys.

In the next sections, we describe rekeying in details.

2.5.1 Supercluster Joining

After receiving the access device, the user does not belong to a supercluster yet. He has only his UK and the KEKs in his path to the root node. After communicating his supercluster to the GM, he will receive the corresponding clusters keys. According the backward secrecy requirement, the GM has to renew all keys of the clusters in the supercluster. For each cluster, the GM has to provide the new cluster key separately to the cluster old members and new member. Let us suppose that u_x joins the supercluster s_y. Then, GM sends the following rekeying messages:

$$\forall p_i \in s_y \begin{cases} GM \rightarrow (u_x) : & \left\{ \overline{K_{p_i}} \right\}_{k_{u_x}} \\ GM \rightarrow (all) : & \left\{ \overline{K_{p_i}} \right\}_{K_{p_i}} \end{cases}$$

With reference to Figure 2.4, when u_7 subscribed $s_y = \{p_2, p_3\}$, he sent the request to GM, then the GM computed $\overline{K_{p_2}}$ and $\overline{K_{p_3}}$. Finally, the GM sent the following messages:

$$\begin{aligned} GM \rightarrow (u_7) &: \left\{ \overline{K_{p_2}} \right\}_{k_{u_7}} \\ GM \rightarrow (all) &: \left\{ \overline{K_{p_2}} \right\}_{K_{p_2}} \\ GM \rightarrow (u_7) &: \left\{ \overline{K_{p_3}} \right\}_{k_{u_7}} \\ GM \rightarrow (all) &: \left\{ \overline{K_{p_3}} \right\}_{K_{p_3}} \end{aligned}$$

We underline that the communication cost for each update of cluster key is constant and equal to two rekeying messages.

2.5.2 Supercluster Leaving

Let us suppose that a user want to terminate the relationship with the broadcast service, he has to leave his supercluster. It means that the user has to leave each cluster in his supercluster. The operation is not trivial and it may be quite expensive in terms of communication cost. For sake of clarity, we will describe the case of a supercluster with one cluster and then we will extend the solution to every type of supercluster.

Algorithm 1 Calculate the set of KEKs to leave from cluster

Require: user u leaves cluster p
  $W \Leftarrow \emptyset$
  $L \Leftarrow$ node_in_path(u)
  for all $i \in \{G(p) - u\}$ **do**
    $node \Leftarrow i$
    while $node \notin L$ **do**
      if $node \in$ childs_of(L) **then**
        $W \Leftarrow W + node$
      else
        $node \Leftarrow$ parent_of($node$)
      end if
    end while
  end for
  return W

According with forward secrecy requirement, when a user leaves a cluster, GM must renew the cluster key. As already said, the GM exploits the LKH tree to reduce the communication cost. Thus, GM computes which KEKs are necessary to encrypt the new cluster key. We propose the Algorithm 1 to achieve the task. The algorithm computes the path from the leaving member's leaf to the root and establishes a node set L consists of the nodes along that path. Subsequently, the algorithm travels along path to the root (LKH tree) of each cluster's member and stops itself each time it arrives at a node that is child of a node in L. This set of children nodes, called W, contains the nodes of KEKs that must be used to distribute the new cluster key. By using these keys, we prevent the leaving member from eavesdropping the new cluster key. Furthermore, we must encrypt the new cluster key with the old cluster key in order to assure that only cluster's members can receive it. So, if a user u_x leaves a supercluster with only a cluster p_y, then GM sends the following rekeying messages:

$$GM \rightarrow (all) : \left\{ \left\{ \overline{K_{p_y}} \right\}_{K_{p_y}} \right\}_{k_i} ; \forall i \in W$$

The number of nodes in W is variable. It depends on the distribution of the cluster's members into LKH tree. It is also influenced by the ratio between the number of cluster's members and n. Anyway, the number of node in W is bound between 1 and $\log(n)$.

$$1 \leq |W| \leq \log(n)$$

Extending the procedure to a supercluster with more clusters, we realized that replaying the procedure for each cluster is not efficient solution. Our solution consists in creating a temporary set of all members of all clusters involved in the rekeying operation exclusive of the leaving member. Exploiting the temporary set, the GM issues the new cluster keys encrypted with the old ones.

Algorithm 2 Calculate the set of KEKs to issue K_{tmp}

Require: user u leaves supercluster s
  $W \Leftarrow \emptyset$
  $L \Leftarrow$ node_in_path(u)
  for all $i \in \{G(s) - u\}$ **do**
    $node \Leftarrow i$
    while $node \notin L$ **do**
      if $node \in$ childs_of(L) **then**
        $W \Leftarrow W + node$
      else
        $node \Leftarrow$ parent_of($node$)
      end if
    end while
  end for
  return W

The Algorithm 2 is the extension of Algorithm 1 to the supercluster.

The GM uses the Algorithm 2 to identify the KEKs set (W). The W set allows to issue K_{tmp} to all members of all clusters involved in the rekeying operation exclusive of the leaving member. Thus, GM generates K_{tmp} and sends the following messages:

$$GM \rightarrow (all) : \{K_{tmp}\}_{k_i} ; \forall i \in W$$

Now, the GM can communicate exclusively with the remaining members of all clusters involved. In order to renew the old clusters keys, the GM sends the following rekeying messages:

$$GM \rightarrow (all) : \left\{ \left\{ \overline{K_{p_i}} \right\}_{K_{p_i}} \right\}_{K_{tmp}} ; \forall p_i \in s$$

With reference to Figure 2.4, let us say that u_6 want to leave his supercluster $s_y = \{p_2, p_3\}$. The GM has to renew K_{p_2} and K_{p_3}. Thus, the GM starts the Argorithm 2 to identify the set $W = \{k_2, k_6\}$. The GM generates K_{tmp} and broadcasts the following messages:

$$GM \rightarrow (all) : \{K_{tmp}\}_{k_2}$$
$$GM \rightarrow (all) : \{K_{tmp}\}_{k_6}$$

Note that users, which do not belong to p_2 or p_6, receive the K_{tmp}. This is not a problem for the security of the rekeying operation because, in the next step, only the users which know the old clusters keys can receive the new ones. The rekeying procedure finishes with the following messages:

$$GM \rightarrow (all) : \left\{ \{\overline{K_{p_2}}\}_{K_{p_2}} \right\}_{K_{tmp}}$$
$$GM \rightarrow (all) : \left\{ \{\overline{K_{p_6}}\}_{K_{p_6}} \right\}_{K_{tmp}}$$

2.5.3 Supercluster Switching

Let us suppose that a user want to change his supercluster, for instance he want to add or remove some contents. This is a very common operation in a contents distribution system. The GM has to identify the initial and final superclusters which we call respectively s_{start} and s_{end}. Then, the GM identifies three sets of clusters: the set of all clusters that the user joins ($s_J \equiv s_{end} \backslash s_{start}$), the set of all clusters that the user leaves ($s_L \equiv s_{start} \backslash s_{end}$) and the set of clusters that belong to both s_{start} and s_{end} ($s_{start} \cap s_{end}$). The clusters into the last set do not need to renew the clusters keys because their membership list does not change. To perform the switch operation, the GM acts only on s_J and s_L.

Using the two operations above described, the GM performs a supercluster joining on s_J and a supercluster leaving on s_L. Even if it may occur that s_J or s_L is empty, the supercluster switching is an expensive operations.

2.6 Security Analysis

The main requirement of a GKM scheme is *confidentiality*, it means that only valid users should be able to decrypt the multicast data, even if the data are accessible to all user in the network. This requirement can be translated into three requirements on key distribution. **Nongroup Confidentiality** asserts that passive adversaries which were never part of the group should not have access to any group key. In our case, the groups are clusters and the GM encrypts every cluster key update, so that a passive

adversary cannot get the cluster key without knowing the dencryption key. Hence, MG-LKH satisfies this requirement.

Forward secrecy claims that users evicted from the group do not have access to any future group key used to encrypt data. According to the procedure in Section 2.5.2, a leaving user is shut out from the rekeying procedure, so that he cannot get the new cluster key. Thus, the forward secresy is assured.

Backward secrecy claims that a user added to a group should not have access to any group key used before his joining. According to the procedure in Section 2.5.1, a joining user should know a previous cluster key to obtain a past cluster key, so that he cannot obtain any past cluster key. Thus, the backward secresy is assured.

2.7 Perfoemance Analysis

In this section, we illustrate the communication, storage and computational cost of MG-LKH. First, we made an analitic analysis on MG-LKH, after that we tested the scheme by means of simulations to characterize some parameters which depend on the graph topology and the users distribution.

2.7.1 Communication cost

We first analyse the performance of MG-LKH in terms of number of message for each rekeying operation. We neglected a dettailed analysis on *service joining/leaving* operations because their communication cost is equal to the joining/leaving oparations of LKH. Their cost is $2log(n)$ messages, where n is the number of users of the broadcast service. We focussed on the supercluster operations. These considerations are valid also for the next analyses.

In this analysis, we define $|s_x|$ the number of clusters associated to s_x and N_{msg} is the number of messages for each rekeying operations.

Supercluster joining

Let us suppose that a user joins a supercluster s_x, thus he joins all the clusters belonging to s_x. As shown in Section 2.5.1, the number of messages to join a cluster is costant and equal to two. Hence, the number of messages depends only on the number of clusters in s_x.

$$N_{msg} = 2|s_x|$$

Supercluster leaving

Let us suppose that a user leaves a supercluster s_x, thus he leaves al the clusters belonging to s_x. In this case, the communication cost dependes on two parameters: the number of clusters in s_x and $|W|$ that is the size of set W (see Section 2.5.2). $|W|$ is bounded between 1 and $log(n)$ and its value depends on the graph topology and the users distribution.

$$N_{msg} = \begin{cases} |W| & |s_x| = 1 \\ |W| + |s_x| & |s_x| > 1 \end{cases}$$

Supercluster switching

Let us suppose that a user leaves the supercluster s_{start} and joins the supercluster s_{end}. As described in Section 2.5.3, the GM computes s_J and s_L. The number of message are the sum of a supercluster joining and a supercluster leaving.

$$N_{msg} = \begin{cases} 2|s_J| + |W| & |s_L| = 1 \\ 2|s_J| + |W| + |s_L| & |s_L| > 1 \end{cases}$$

2.7.2 Computation cost

The computation cost is bound to the number of operations of encryption and decryption. Moreover, the computation costs have different values if we consider the cost on GM side or user side.

We define C_E the everage computational cost of a encryption/decryption operation and C_r is the computational cost of generating one key from a cryptographically-secure random source.

Table 2.1 shows the computation cost for each user involved in the rekeying operation. All the values depend only on the number of clusters into the supercluster and they do not depend on the number of users. Whereas the GM has to send at the very least an information for each cluster involved, these results are close to the lower bound reachable in this case.

Table 2.2 shows that the computation costs on the GM side depend on the number of clusters involved plus a logarithmic factor on the number of users.

2.7.3 Storage cost

For the sake of completeness, we analysed the storage cost even if it is a secondary issue. Due to the costant growth in the storage thecnologies, nowadays it is easy that a device can hold thousands of keys. k is the size in byte of a key, $|s_x|$ is the number of program associated to s_x and $|P|$ is the number of all programs.

Table 2.1. User side computetion costs (SJ is supercluster joining, SL is supercluster leaving and SS is supercluster switching).

	User x
SJ	$C_E\|s_x\|$
SL	$\|s_x\| = 1 \rightarrow 2C_E$ $\|s_x\| > 1 \rightarrow C_E(2\|s_x\| + 1)$
SS	$\|s_L\| = 1 \rightarrow C_E(\|s_J\| + 2)$ $\|s_L\| > 1 \rightarrow C_E(\|s_J\| + 2\|s_L\| + 1)$

Table 2.2. GM side computetion costs (SJ is supercluster joining, SL is supercluster leaving and SS is supercluster switching).

	GM
SJ	$(2C_E + C_r)\|s_x\|$
SL	$\|s_x\| = 1 \rightarrow C_r + 2C_E\|W\|$ $\|s_x\| > 1 \rightarrow C_r(\|s_x\| + 1) + C_E(\|W\| + 2\|s_x\|)$
SS	$\|s_L\| = 1 \rightarrow C_r(\|s_J\| + 1) + 2C_E(\|W\| + 1)$ $\|s_L\| > 1 \rightarrow C_r(\|s_J\| + \|s_L\| + 1) + C_E(\|W\| + 2(\|s_J\| + \|s_L\|))$

Table 2.3. Storage cost.

User x	GM
$k(\lceil log(n) \rceil + \|s_x\| + 1)$	$k(2n + \|P\| + 1)$

2.7.4 Simulations

Table 2.4. CTest cases.

Case	Major superclusters	Major events
Case 1	Multiple	Join and leave
Case 2	Single	Join and leave
Case 3	Multiple	Switch
Case 4	Single	Switch

In the simulations, we compare MG-LKH with KTR that showed to have better performance than other schemes like SKT, eLKH and LKH. The two schemes work on the some cluster/supercluster model. Thus, we performed the simulations with the same set up in [25] so that it makes possible compare the two GKM schemes. The set up assumes that the service provides 50 clusters and 300 different options of superclusters. There are 10,000 users (on the everage) subscribing to the services.

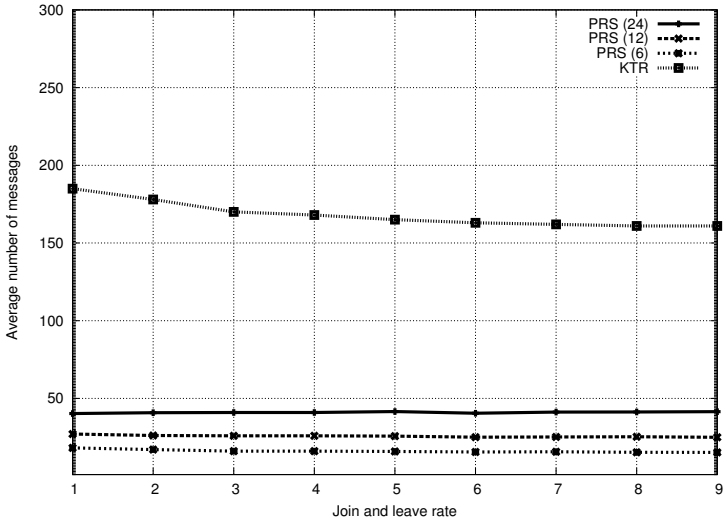

Figure 2.5. Case 1. Average rekey message size of KTR and MG-LKH .

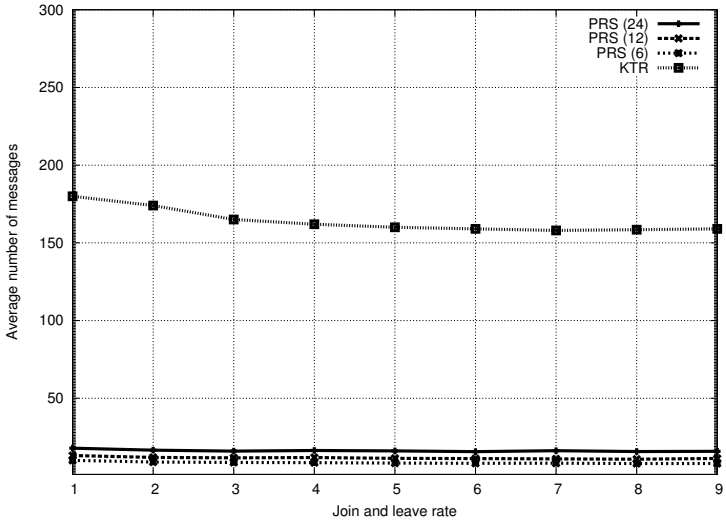

Figure 2.6. Case 2. Average rekey message size of KTR and MG-LKH .

The LKH tree with 10,000 users is automatically generated and it has a depth of 14, since $2^{14} > 10,000$.

The user events are supercluster joining, leaving and switching. They are modelled as independent Poisson processe with $\lambda_l = \lambda_j$, so that the total number of users remains constant. As in [25], we vary λ_l and λ_s in order to observe how the rekey performance changes. The simulation performs 3,000 random user events and computes the average rekeying cost.

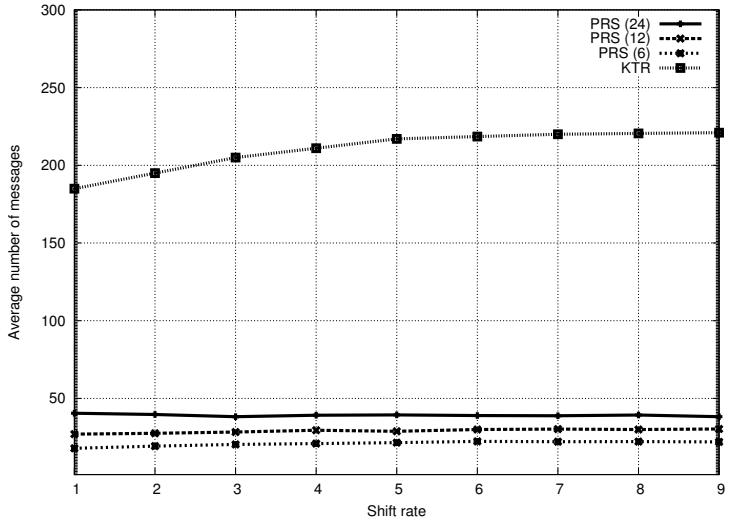

Figure 2.7. Case 3. Average rekey message size of KTR and MG-LKH .

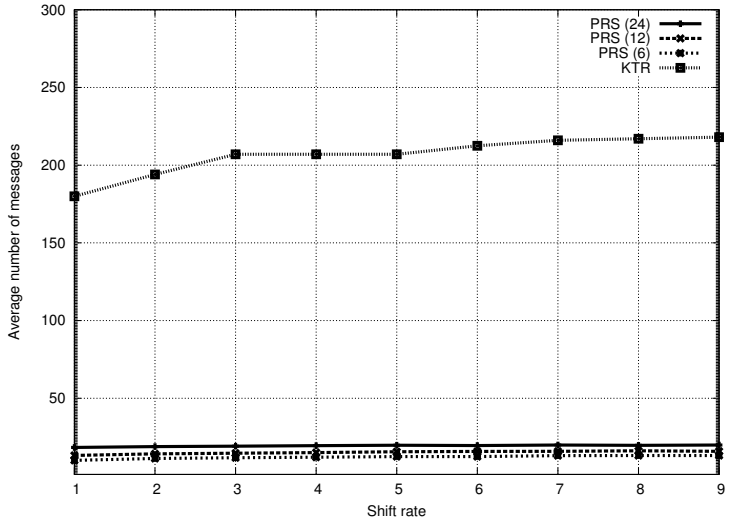

Figure 2.8. Case 4. Average rekey message size of KTR and MG-LKH .

The simulations analyze four test cases which are different in terms of major superclusters and major events, as shown in Table 2.4. In Cases 1 and 3, 20 percent of users subscribe to one cluster and 80 percent of users subscribe to multiple clusters, vice versa in Cases 2 and 4. With respect to major events, Case 1 and 2 are with predominance of join and leave and Case 3 and 4 are with precominance of switch. In each Case, the rates for the major events vary from 1 to 9 while keeping the other rates at 1.

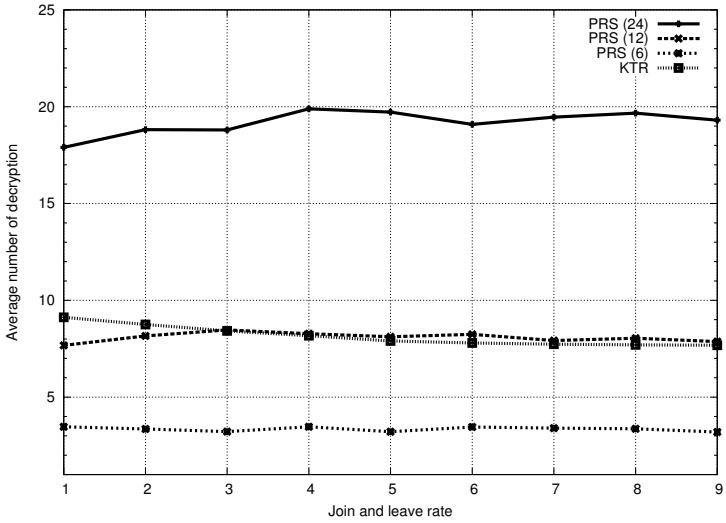

Figure 2.9. Case 1. Average number of decryptions per user per event.

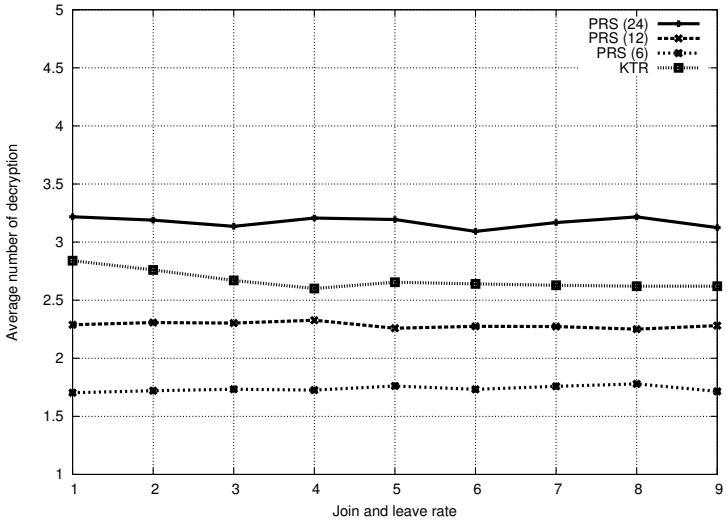

Figure 2.10. Case 2. Average number of decryptions per user per event.

We analyze two performance metrics. The *average rekey message size per event* is the numbero of keys sent in the rekey message and represents communication cost. The *average number of decryption per event per user* measures the computation cost for each user, this result is closely related to power consumption in a mobile device.

We have focussed on these two metrics because they depend on $|W|$ that is a variable value between 1 and $\log(n)$. $|W|$ depends on the topology of the tree that changes over the time due to the sequence of user events.

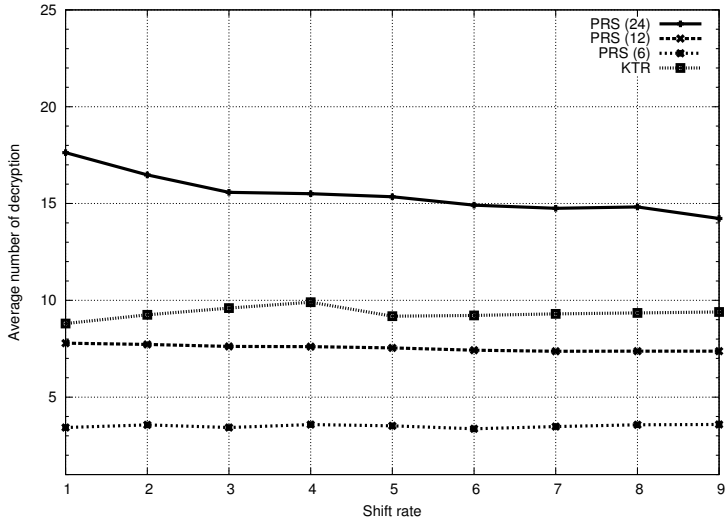

Figure 2.11. Case 3. Average number of decryptions per user per event.

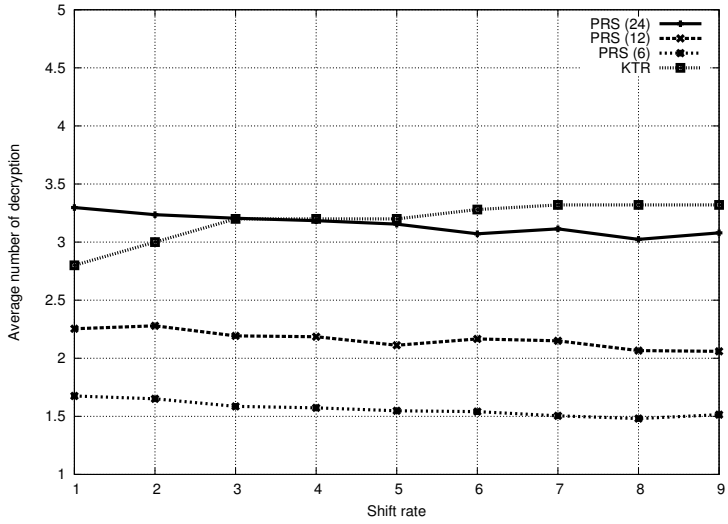

Figure 2.12. Case 4. Average number of decryptions per user per event.

Moreover, we have tested MG-LKH varying the average number of clusters per supercluster. We have analyzed three cases with an increasing average number of clusters per supercluster, that is 6, 12 and 24.

The charts in Figures 2.5, 2.6, 2.7 and 2.8 show the average rekey message size per event for each test case in Table 2.4. It is evident that MG-LKH has a much better performance, the communication overhead is considerably cut down. The communication is the bottleneck of the performance of a rekeying scheme, thus reducing

by one order of magnitude the message size per event is a relevant result. More in depth, the performance of MG-LKH decreases when the average number of clusters per supercluster increases, as we expected. The average number of clusters per supercluster weighes on the values of $|s_x|$, $|s_J|$ and $|s_L|$ (see Section 2.7.1). From these results, our conclusions are that it is possible to reduce the communication overhead keeping the same number of clusters. Otherwise, we can increase the number of clusters, providing many more services to the final user, with the same level of communication overhead.

The charts in Figures 2.9, 2.10, 2.11 and 2.12 show the average number of decryptions per user per event. In this analysis, the difference between the two algorithms is less marked. In any cases, the performance of MG-LKH get worse with the growth of the everage number of clusters per supercluster. This beaevior is explained in Section 2.7.2. The average number of clusters per supercluster weighes on the values of $|s_x|$, $|s_J|$ and $|s_L|$. Anyway, only the results of MG-LKH with an average of 24 clusters per supercluster have comparable or worse performance than KTR. However, during these last years, the problem of computation cost has lost importance, because the constant growth of the computing power of modern CPUs and their high level of integration allow a pocket-sized device to have enough resources to achieve complex cryptographic operations.

To conclude our analysis, we have investigated on the behavior of parameter $|W|$ because it influences the communication cost (Section 2.7.1). The value of $|W|$ is variable for each leave/switch event, it is bound between 1 and $\log(n)$ (see Section 2.5.2). We have simulated 6,000 random user events in a group of 10,000 of users varying the ratio between the everage number of user belonging to a cluster and the number of members of whole service: $|G(p_i)|/n$.

The chart in Figure 2.13 shows the probability distribution of parameter $|W|$. With $|G(p_i)|/n = 10\%$, the great majority of events need a $|W|$ between 9 and 11 over 15 that is the maximun value of $|W|$. It means that MG-LKH saves the 33% of rekeying message in a supercluster leaving operation. Ingreasing the ratio $|G(p_i)|/n$, the efficiency of MG-LKH decreases as shown in Figure 2.13. In the worst case ($|G(p_i)|/n = 90\%$), that is the number of all members belonging to a cluster is comparable to the whole service group, the great majority of events need a $|W| = 13$. In other words, MG-LKH saves the 15% of rekeing message. Moreover, these results are relevant in terms of computation cost on GM side. As shown in Table 2.2, the computation cost of supercluster leaving/switching events depends on $|W|$.

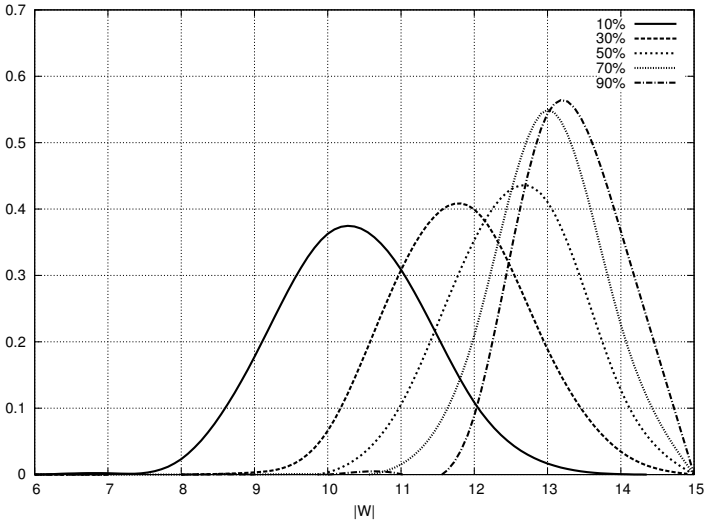

Figure 2.13. Probability distribution of the parameter $|W|$.

3

Reliability

Multi-vehicle systems composed of Autonomous Guided Vehicles (AGVs) are largely used for industrial transportation in manufacturing and logistics systems as they offer potential advantages with respect to single-agent systems in terms of task speedup, robustness and scalability. For instance, a typical function of a multi-AGVs system consists in transporting raw or semi-finished material from warehouse to production lines [1]. However, deployment of a team of AGVs raises management and coordination problems such as collision avoidance, conflict resolution, and shared resources negotiation [1].

Coordination of a team of AGVs can be either centralized or distributed. In the majority of industrial application, AGVs coordination is centralized where a single decision maker is responsible for solving task allocation, motion planning, and coordination problems. A centralized solution is easier to implement but the decision maker becomes a performance bottleneck with severe limitations in terms of scalability.

Decentralized approaches are more suitable than centralized ones for dealing with coordination problems involving a large number of AGVs [49]. These approaches are divided into two phases, the planning phase where paths are planned using independent objectives for each AGV, and the coordination phase where coordination and interaction of each AGV with the other AGVs take place. Scalability is achieved because in the coordination phase each AGV takes decisions that require only the knowledge of the state (e.g., position and speed) of its neighbors.

In a decentralized approach reliable and secure dissemination of the state is a crucial operation that is often neglected. If dissemination is unreliable, neighboring AGVs may achieve an inconsistent view of the system leading to the failure of the coordination task. This has clear safety implications. If dissemination is not secure, an adversary may modify or inject fake messages so leading AGVs to achieve wrong and/or inconsistent views. Once again this may cause the coordination task to fail.

In this chapter we focus on reliable state information exchange among neighbors and present an efficient and scalable mechanism for neighborhood monitoring. Intuitively, this *Neighborhood Monitoring Protocol* (NMP) is crucial for reliable state dissemination because, when an AGV broadcasts its state, an accurate and timely notion of its neighborhood allows it to track which neighbors have received such state and which have not and thus need a re-transmission.

The protocol operates over an IEEE 802.11 wireless network technology [24]. This technology is rapidly expanding in industrial scenarios due to its recent improvements in terms of hardware costs, transmission speed, and simplicity and flexibility of deployment [13]. However, it lacks any reliable broadcast service.

In the literature there are many solutions that strive to improve the reliability of 802.11 broadcast communication at both the Medium Access Control (MAC) layer [57, 58, 55, 52, 32] and upper layers [2]. NMP operates above the MAC layer because we are not interested in improving the reliability of the whole broadcast traffic. Rather, we are interested in increasing the reliability of the state dissemination aimed at AGVs coordination service without influencing the broadcast traffic of other protocols. Therefore, starting from the periodic nature of the state dissemination traffic pattern, we have designed an accurate, efficient, and scalable protocol that is suitable for real-time coordination protocols. NMP is accurate because the difference between the actual neighborhood of an AGV and the view of that neighborhood the protocol provides the AGV is negligible. Moreover NMP makes it possible to estimate a maximum state dissemination delay that is fundamental in real-time applications. Furthermore, NMP is efficient as it produces a negligible rate of collisions and consequent packet loss so solving the insidious problem that afflicts the periodic and uncoordinated data dissemination protocol, namely the overlapped transmissions. Differently from other approaches, NMP is scalable because state dissemination influences only actual neighbors and the dissemination rate is automatically reduced when neighbors are absent.

3.1 Related Works

The IEEE 802.11 broadcast protocol is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and does not offer any recovery mechanisms for broadcast frames [24]. In 802.11, the Distributed Coordination Function (DCF) incorporates CSMA/CA and acknowledgement (ACK) and is the principal access method to share the wireless channel. Optionally, the mobile devices can make use of the virtual carrier sense mechanism for unicast transmissions by means of Request To

Send (RTS) and Clear To Send (CTS) control frames to eliminate the hidden terminal problem. For broadcast packet, 802.11 devices cannot exploit the RTS/CTS/ACK mechanisms. Hence they simply execute CSMA/CA and broadcast data frame incur in an increased probability of getting lost due to collisions. Much work has been done to improve the reliability of IEEE 802.11 broadcast. Many of the proposed solutions face with the problem at MAC layer.

K. Tang and M. Gerla propose Broadcast Support Multiple Access (BSMA) that extends the use of RTS/CTS/NACK control frames to broadcast communication [57]. Before data transmission, a sender broadcasts an RTS frame to its neighbors. Consequently, neighbors reply with a CTS. Upon receiving a CTS frame from any neighbor of its, the sender broadcasts the DATA frame. If all neighbors correctly receive the DATA frame then the transmission ends. Otherwise, the neighbors that have not correctly received the DATA frame send a NACK frame to have the sender retransmit the DATA frame. BSMA has a weakness. Actually, BSMA is not able to coordinate transmission of CTS or NACK frames. Thus collisions of CTS or NACK frames may occur. A failure in receiving CTS frames makes the sender to re-transmit the RTS frame with negative effects in terms of efficiency.

The Broadcast Medium Window (BMW) follows a different approach [58]. A sender broadcasts a DATA frame by unicasting it to every neighbor. Reliability is achieved by means of the RTS/CTS/DATA/ACK mechanisms. As it turns out, reliability of this scheme is improved at the cost of lowered efficiency.

The Batch-Mode Multicast MAC (BMMM) introduces a new additional control frame called Request for ACK (RAK) [55]. After RTS/CTS, a sender broadcasts a DATA frame and then a RAK frame to coordinate ACK frames from receivers. The main drawback of this approach is that if an ACK is lost, the sender has to restart the whole CTS/RTS/DATA/RAK/ACK procedure.

Taking into account that the broadcast unreliability is mainly caused by the lack of acknowledgement of broadcast frames, the approach proposed by Sheu et al. suggests to make use of ACK frame after the broadcast DATA frame transmission [52]. Sheu et al. consider a period of time after DATA transmission that is called Back-off ACK (BACK) and defined as the time interval between the end of the SIFS and the end of the DIFS during which the sender receives the ACKs. The BACK window is divided into mini-slots and the receivers coordinate the ACK transmission choosing randomly one mini-slot to transmit an ACK. This solution has an evident drawback, namely, if one ACK is missed the sender will restart the whole procedure.

Another proposed solution is Multiple Access Collision Avoidance protocol for Multicast services (MACAM) [32]. This solution includes a list of nodes (neighborhood) in the RTS frame. Neighbors respond sending CTS frames according to the sequence

of their addresses specified in the RTS. This mechanism increases the length of the CTS control frame. Moreover, if the number of neighbors is greater than the maximum number of recipients that can be included in the RTS, the sender must perform multiple DATA transmission. Consequently, reliability increases but efficiency decreases.

All the solutions presented so far attempt to achieve broadcast reliability at the MAC layer. Another possible approach addresses the problem at the upper layer. This approach is adopted by the Real-Time Data Base (RTDB) middleware [2]. RTDB provides an efficient and timely support for the fusion of distributed perception and the development of coordinated behaviors by means of a distributed database that is partially replicated in all involved devices. The database contains local and remote state variables that are updated periodically and automatically in the background by the dissemination of multicast packets at a refresh rate that is adapted to the data dynamics. To reduce access collision among communicating agents, RTBD uses an adaptive Time Division Multiple Access (TDMA) transmission control mechanisms, with a predefined round period called team update period, so they have to use a base station to manage the communication. This solution solves the problem of collisions between broadcast transmissions. However, each device has to wait for all other devices until they finish their transmissions. Moreover, this approach cannot be used in a Mobile Ad-hoc NETwork (MANET) scenario because there is not a coordination point that manages the TDMA.

Given the inefficiencies of current protocols (increased number of frames, increased number of collisions, increased data transmission time), and the requirements of reliable state dissemination (scalability, accuracy, and predictable dissemination delay), we propose a new approach that makes an efficient use of wireless medium. The proposed approach is at the application layer, exploits the periodic nature of the state dissemination protocol, and is suited for a MANET of AGVs.

3.2 System Model

We consider a system composed of a set of AGVs that share a common environment to fulfill their task either in isolation or in group. Vehicles cooperate at least for collision and deadlock avoidance. As vehicles share a common environment, collision avoidance prevents any vehicle from colliding into another. At the same time, deadlock avoidance prevents a sub set of vehicles from stalling because they are not able to solve conflicts possibly leading to collisions. In order to cooperate for collision and deadlock avoidance, vehicles periodically disseminate their respective state (position and speed) through a wireless ad-hoc network. We focus on IEEE 802.11 but the following arguments can be applied to other wireless technologies too.

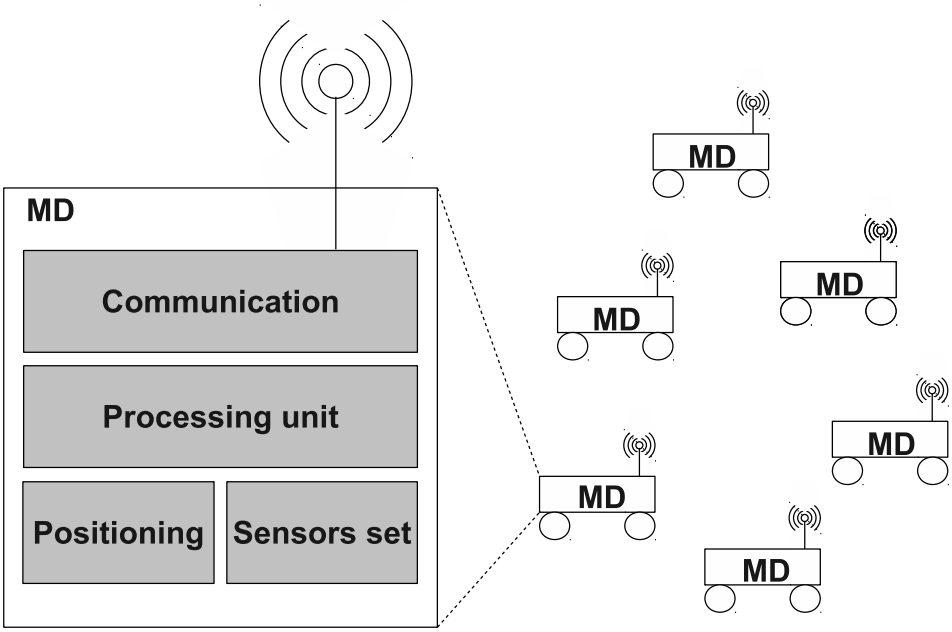

Figure 3.1. The Mobile Devices (MD) share the same environment.

Intuitively, each vehicle detects its neighbors and, contextually, disseminates its state to such neighbors as follows. Every vehicle periodically broadcasts a STATE packet that coveys both mobility information and possible sensor readings (e.g., temperature). At the same time, the vehicle receives the STATE packet broadcast by neighboring vehicles, stores the most recent ones in an internal buffer, and makes them available to applications. So doing the vehicle is able to keep track of position and speed of its neighbors and then use these information items in solving coordination problems. Neighborhood discovery and state dissemination must be reliable or, otherwise, neighboring vehicles may achieve an inconsistent view of the system. For instance a vehicle may miss the presence of another one or believe that this one is in a different position. These inconsistencies in the view may cause the vehicle to take maneuvers that are inconsistent with those taken by its neighbors and thus conducive of possible safety consequences.

In coordination problems, the notion of neighborhood is application-specific and defined on geographical basis. We call *neighbors* any two vehicles whose distance is smaller than, or equal to, the *neighborhood distance* D_n. Let D_c be the *communication radius*, i.e., the maximum distance allowing communication between two AGVs.

Of course, it must be $D_n < D_c$, because otherwise no communication among neighbors would be possible. Except for this, D_n must be large in order to detect a new neighbor before it is too close. For instance, in a collision avoidance protocol, the largest D_n the smallest the risk of collision because a neighbor is discovered when it is still far "enough". On the other hand, D_n must be not "too" large in order to consider neighbors only the vehicles actually necessary for the coordination protocol. More generally D_n depends on specific application. In this paper, we assume that D_n is defined at system initialization and never changes afterwards.

We assume that a vehicle is able to localize itself. Localization technologies are available for both outdoor (e.g., the Global Positioning System) and indoor environments [34], see also Chapter 4. Furthermore, we assume that clocks of mobile devices are synchronized. The commonest solutions are the Network Time Protocol (NTP) and its variations [36], [37]. Finally, we assume that vehicles are equipped with a set of sensors to sense the surrounding environment. Figure 3.1 summarizes briefly the system model.

3.3 The Neighborhood Monitoring Protocol

The objective of NMP is to allow a vehicle to reliably detect and track the vehicles that belong to its neighborhood. To accomplish that, a vehicle periodically transmits its state with a given *state transmission frequency* F. More precisely, a vehicle broadcasts a STATE packet which conveys the current vehicle *position*, *speed* and *sensor readings*.

In the STATE packet, the vehicle also inserts the timestamp t_i, i.e., the instant of the STATE packet creation, and ΔT, i.e., the time interval before the next STATE packet. So $t_{i+1} = t_i + \Delta T$ specifies the time when the next STATE packet will be sent. Upon receiving a STATE packet from n_i, a neighbor n_j can determine the next packet time $\tau_i = t_{i+1} + \Delta T_{tx}$ where ΔT_{tx} is an estimation of the transmission delay. Thus, if the next packet from n_i does not arrive by τ_i, the mobile node n_j considers it as lost and sends a NACK packet to stimulate the retransmission of a new STATE packet by n_i.

It is worthwhile to highlight soon an important difference between NMP and existing solutions such as BMMM [55] and that based on Sheu's et al.'s protocol [52]. These protocols need to notify each time they receive a data packet because the protocol does not know the data traffic behaviour and thus they cannot foresee when the next packet will arrive. In contrast, NMP can establish when the next STATE packet will arrive and thus can identify when it is missed. Consequently, NMP need to notify only error conditions so reducing the communication overhead.

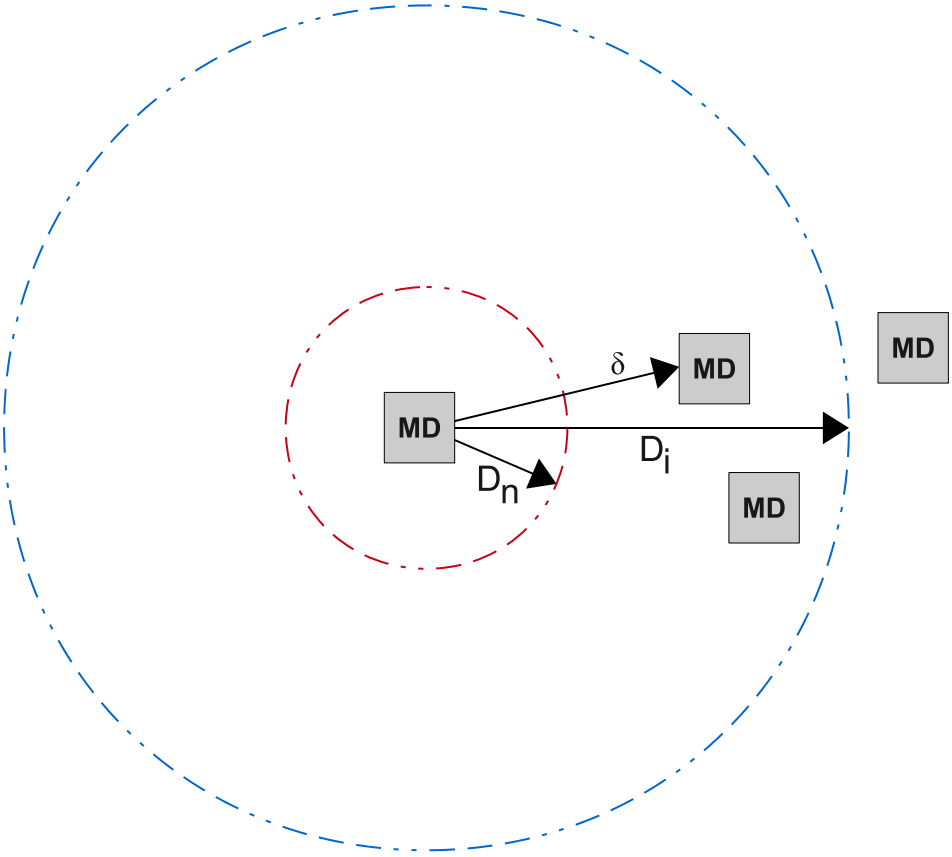

Figure 3.2. The distance parameters D_n and D_i.

Of course the NACK mechanism of NMP might cause several collisions because, in the case a STATE packet is lost, two or more neighbors might attempt to transmit a NACK at the same time. In order to solve this problem, we take advantage of the distributed nature of the problem of coordinating the NACK transmissions. When the deadline for the next STATE packet expires, the mobile device activates a back-off window composed of eight fixed slots. Then, the mobile device selects one of these slots at random and transmits the NACK packet in such a slot. So doing, the probability that two or more mobile devices collide when sending a NACK becomes negligible. If the back-off window expires and the data retransmission does not occur, mobile devices double the back-off window and repeat the NACK transmission process. Every node can transmit a NACK for four times. After that the node removes the not responding node from its neighborhood. On the contrary, if the STATE retransmission occurs, the recipient mobile device stops the back-off timer. Summarizing the coordination operations, the protocol waits for a STATE packet. If it is missed (timer expired) the protocol

sends a NACK after a back-off time and returns to wait for a new STATE packet. If the missed STATE packet arrives during the back-off time then the protocol stops sending the NACK.

This retransmission procedure solves a insidious problem that afflicts periodical updates in uncoordinated protocols, namely, the *overlapped transmissions*. Consider a random initialization scenario with a large number of mobile devices. In such scenario, it is highly probable that two or more devices may want to transmit at the same time. This increases the likelihood of collision. With a constant update period, the problem occurs at each transmission and consequently it may cause a considerable packets loss. The retransmission procedure slightly delays the packet and spreads the overlapped transmissions so the overlapping will not occur in future.

Moreover, NMP has the ability to reduce the number of transmissions when they are not necessary. More specifically, the state transmission frequency F varies between two values F_{min} and F_{max}, $F_{min} \leq F_{max}$, in a way inversely proportional to the distance of the closest vehicle within a fixed distance called idle distance D_i with $D_c > D_i > D_n$ (see Figure 3.2).

If the closer neighbor is between D_n and D_i, device vehicle does not keep track of that neighbor state but uses its position information in order to vary the transmission frequency F. Hence the area between D_n and D_i is a safety zone whose task consists in adapting F to F_{max} when the closest neighbor gets near to the vehicle. F_{max} is set according to the application requirements. For instance, the transmission frequency of position information in a coordination protocol for the collision avoidance follows an empiric formula $F_{max} > 2V_{max}/D_n$ where V_{max} is the maximum vehicle speed. Intuitively, the formula says that during two consecutive transmissions two vehicles cannot get closer than D_n.

Differently, F_{min} aims at reducing the access to the shared wireless and thus reduce the network load. With reference to Figure 3.2, if we denote by δ the distance of the closest neighboring node, the value of the state transmission frequency F is given by

$$F = \begin{cases} F_{min} & \text{if } \delta > D_i \\ F_{min} + (F_{max} - F_{min})(D_i - \delta) over D_i - D_n & \text{if } D_n \leq \delta \leq D_i \\ F_{max} & \text{if } \delta > D_i \end{cases} \tag{3.1}$$

F_{min} and D_i values must be configured carefully, taking into account the mobility parameters of vehicles (e.g., the speed). Wrong values for F_{min} and D_i may cause the presence of vehicles within D_n while the state transmission frequency is lower

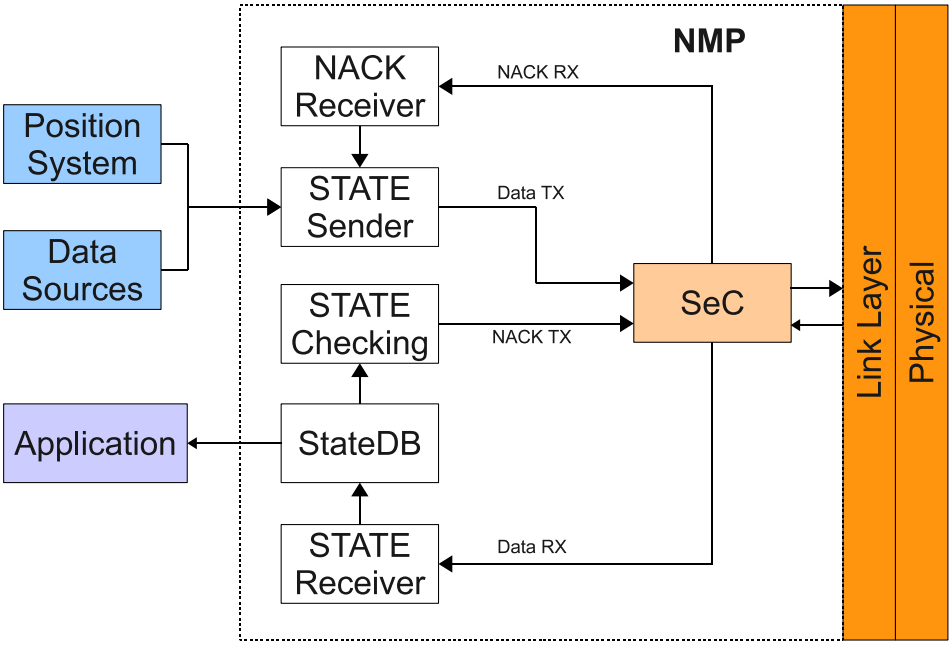

Figure 3.3. NMP software architecture.

than the value F_{max} required by the application. Given D_n and V_{max}, we suggest to choose F_{min} such that $F_{min} \geq 2V_{max}/(D_i - D_n)$ in order to guarantee a correct data frequency when a neighbor gets closer than D_n. According to the previous equation, $(D_i - D_n)$ must be greater than the distance covered during two consecutive transmissions, i.e. V_{max}/F.

3.3.1 Software Architecture

NMP operates over link layer using the standard interface provided by 802.11 MAC layer. The communication follows the producer-consumers model according to which each device regularly broadcasts (produces) its state while the remaining ones receive (consume) the state and update their local structures.

Figure 3.3 shows the NMP software architecture. The *DataDB* module records the the states collected from neighbors and provides it to the Application. The other modules perform operations whose semantics are inferable by the module name.

As described in Section 3.3, the state dissemination protocol is based on two packet types, the STATE and the NACK packets. The STATE packet consists of a an *Header* field followed by a *State* field and an *Auth* field. The Header is composed of six fields. The *ID* field specifies the unique identifier of the sending device. The *ServiceID* field specifies the type of data in the state field (e.g., temperature, alarms).

If a packet conveys no state information, the *ServiceID* filed contains zero and the State field is not present. The field *Coordinates* specifies the current coordinates of the sending mobile device. In our implementation, we use plane coordinates, but it can be extended in order to cope with aerial/submarine mobility. The field *Timestamp* specifies the instant when the packet has been created. The field *Nextmsg* indicates when the next STATE packet is going to be sent. Finally, the field *Length* specifies the size in byte of the State field. The *State* field conveys mobility information (e.g. speed, acceleration) that are established during the configuration phase and they are mandatory into every STATE packet. Finally, the *AUTH* field carries a message authentication code computed by means of a keyed hash function that takes as input the Header, the State field, and a group key shared by all mobile devices in the system. The Auth field assures the data integrity and authenticity of a STATE packet and prevents an external attacker from modifying packets or injecting fake ones. Furthermore, the AUTH field together with the timestamp guarantees the freshness of the STATE packet an prevents replay attacks. About the group key management scheme, you can find more details in Chapter 2 or in the articles [17, 4].

The NACK packet contains only four fields: the sender ID, the ID of the device whose STATE packet was lost, a timestamp, and Auth field.

3.4 Performance Evaluation

In order to evaluate performance, we have simulated NMP by means of Omnet++ and INET Framework. We have realized two NMP versions. The *reliable* version implements NMP exactly as described in Section IV. The *unreliable* version implements NMP without the NACK retransmission mechanism, that is a simple periodic dissemination protocol. The comparison of performance of these two implementations is useful to understand the benefit introduced by NMP.

3.4.1 Factors of Performance Evaluation

Our goal is to evaluate efficiency, accuracy and scalability of NMP. In order to quantify them, we define four factors: the percentage of busy channel, the packet loss, the maximum state dissemination interval, and, finally, the diversity. Simulations of the protocol are aimed at assessing these factors. The *percentage of busy channel* (PBC) is defined as the ratio between the channel busy time due to transmission and the simulation time. PBC provides a measure of how much the protocol exploits the wireless channel and thus gives an indication of the overhead introduced by extra-packets adopted during the retransmission procedure (NACK). Thus, PBC is a mea-

sure of the protocol scalability because if PBC grows linearly with the number of AGVs, notwithstanding the presence of control packets to manage the reliability, the protocol scales well. The *packet loss* is defined as the ratio between the number of lost packets due to collisions and the total number of sent packets. Such a factor depends on the numbers of mobile devices involved in the protocol and fixes a practical upper bound to this number. The protocol design is aimed at reducing the packet loss value in order to increase the number of devices that may be present in the system. PBC and packet loss are related because a small packet loss and a PBC linearly growing with the number of AGVs are symptoms of efficient communications. The *maximum state dissemination interval* is the maximum delay between two consecutive state transmissions received from the same neighbor within the range D_n. This value allows us to estimate the maximum delay introduced by NMP when it works in proximity of channel saturation. Such a maximum delay is crucial for the real-time nature of the system. Finally, the *diversity factor* gives a measure of the accuracy of the protocol. Intuitively, it measures how much the neighborhood returned to a vehicle by NMP is adherent to reality. Let N be the neighborhood of a given vehicle and let $\tilde{N}$ be the neighborhood returned to the vehicle by NMP at the same moment. More formally, the diversity factor D is defined as

$$D = \begin{cases} 1 - \frac{|N \cap \tilde{N}|}{|N \cup \tilde{N}|} & \text{if } N \neq \emptyset \text{ and } \tilde{N} \neq \emptyset \\ 0 & \text{if } N \neq \emptyset \text{ or } \tilde{N} \neq \emptyset \end{cases} \tag{3.2}$$

where $| \star |$ denotes the set size operation. If the protocol provides high accuracy, i.e. the detected neighborhood is very close to the real neighborhood, then $\tilde{N}$ tends to N, and the diversity factor D tends to zero. In contrast, if $\tilde{N}$ differs from N, then $N \cap \tilde{N} \rightarrow \emptyset$ and thus the diversity factor D tends to one. Notice that assuring a maximum value for the data delay and a low value for diversity is relevant for protocol reliability.

3.4.2 Simulation Scenario

The simulation scenario consists in a 150 m $\times$ 200 m rectangular area where a fixed number of vehicles follow random trajectories with constant speed between 1 and 2 mps. D_n and D_i are respectively 25 m and 50 m for all mobile devices and all simulations. The communication module emulates a 802.11g NIC set up in ad-hoc mode with 2 Mbps transmission rate. In the simulations, the vehicles exchange just their positions and thus packets have an empty State field (position coordinates are in the packet header). This is not a lack of generality because if speed or acceleration would be necessary, they would require just a few bytes State field. The resulting

packet size would be comparable to that used in the simulation and thus the simulation results remain valid.

Simulations were conducted for both the reliable and unreliable mode varying the number of vehicles between 20 and 120, and the data frequency F_{max} between 8 Hz and 20 Hz. F_{min} is 0.33Hz for all simulations.

3.4.3 Simulation Results

We start analyzing the PBC because the other results depend on it. In Figures 3.4 and 3.5, the two charts showing that PBC grows linearly with the number of mobile devices until 80% where channel saturation occurs. The channel saturation is an upper bound that fixes the maximum number of AGVs. There are not considerable differences between the reliable and unreliable operating mode. The reliable mode is characterized by a slightly greater PBC (+2%) in proximity of channel saturation that is justified by the NACK transmissions. Before the channel saturation, the number of retransmissions is very low. Therefore the protocol does not incur in unnecessary retransmissions and the overhead introduced by control messages (NACK) is negligible.

Figures 3.6 and 3.7 show the packet loss versus the number of nodes. We can see the PBC effects. Actually, the packet loss increases rapidly in proximity of the channel saturation. But with 60 nodes at 20Hz the unreliable mode suffers for packet loss even if the PBC is around 45% because of overlapped transmissions. The reliable mode mitigates the problem, because the retransmission procedure spreads the overlapped transmissions over a larger interval by efficiently using the wireless channel and obviating future overlapping. This behaviour is particularly evident in the reliable mode at 13.3Hz where, differently from the unreliable mode, packet loss is practically absent with less than 100 nodes.

The charts in Figures 3.8 and 3.9 show the maximum data dissemination interval within D_n versus the number of vehicles. In the unreliable mode, overlapped transmissions make the trend of the maximum delay irregular. This makes it hardly predictable. In contrast, thanks to its ability of spreading overlapped transmissions, in the reliable mode the maximum data interval increases slowly and linearly with the number of mobile devices. At 20Hz frequency and 60 nodes, the max data interval is twice the fixed interval at F_{max}.

A PBC greater than 80% causes a quick performance degradation, especially in terms of packet loss due to channel saturation. Therefore, we suggest to operate in the reliable mode, with a PBC value lower than 60% because the packet loss is negligible (under 5%). However, in reliable mode the maximum data interval remains predictable also when we are close the channel saturation. We can establish a delay upper bound also PBC is 90%.

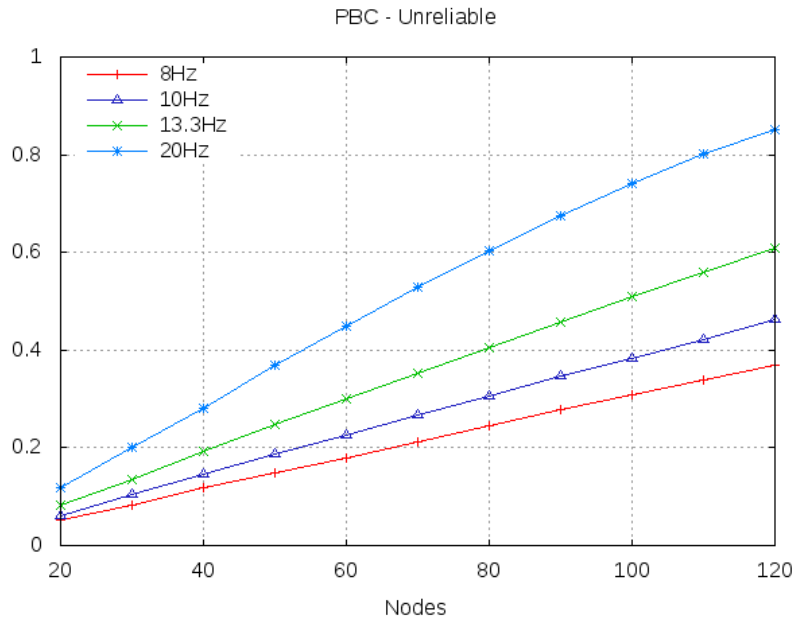

Figure 3.4. Percentage of Busy Channel without NMP.

The value of the diversity factor is always less than 5% for every number of nodes and for every frequency in the considered ranges in both the unreliable and reliable mode. This means that the view of the neighborhood returned by NMP to a vehicle practically coincides with the real neighborhood of that vehicle. This proves the accuracy of our.

To summarize, the reliable mode efficiently uses the wireless channel thanks to the retransmission procedure and the spreading of overlapped transmissions. The PBC charts show that NACKs do not introduce significant overhead, but reduce packet loss and stabilize the maximum data interval. These benefits translate into the possibility of accommodating a larger number of mobile devices. This increases system scalability and guarantees a maximum data delay that increases the data accuracy.

3.5 A Case-Study

We have evaluated NMP with a specific decentralized collision avoidance algorithm based on the Generalized Roundabout Policy (GRP) proposed in [42]. The proposed case-study propose an interesting challenge since it takes into account non trivial agents such as vehicles that move with constant non null velocity (non-holonomic).

The impossibility of stopping the vehicle in case of a conflict makes the communication infrastructure playing a fundamental role to ensure the safety of the system.

The case-study has been evaluated from several points of view by simulation. We show that NMP maintains a high level of delivery ratio, restoring sporadic packets collisions and packets loss that could lead the collision avoidance algorithm to dangerous state as violations of safety conditions. Finally, we have investigated the NMP behaviour even in exceptionally adverse communication conditions and we have compared its performance with a simple periodic dissemination protocol without reliability features.

3.5.1 Collision Avoidance Strategy

We consider the following kinematic model for each agent involved in the system:

$$(\dot{x},\, \dot{y},\, \dot{\theta}) = (u\cos\theta,\, u\sin\theta,\, \omega), \tag{3.3}$$

where u and ω are linear and angular velocity respectively. The linear velocity is supposed to be constant but non zero for any agent. A bound on angular velocity is obtained as $|\omega| \leq \frac{u}{R^c}$ where R^c defines the minimum curvature radius achievable.

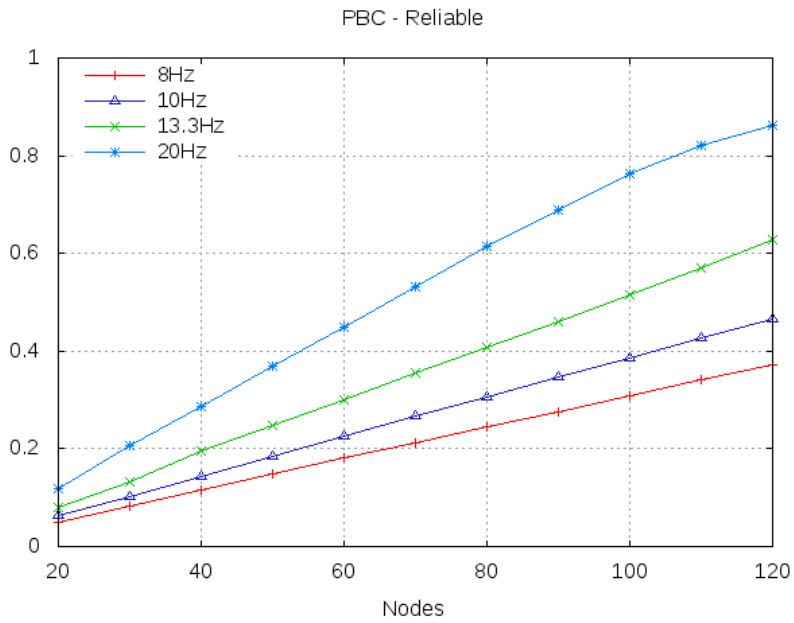

Figure 3.5. Percentage of Busy Channel using NMP.

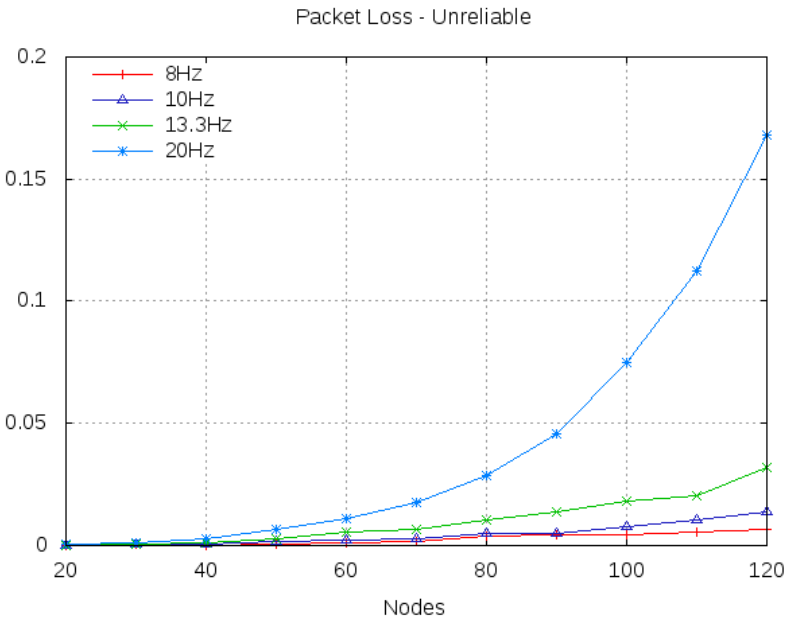

Figure 3.6. Packet Loss without NMP.

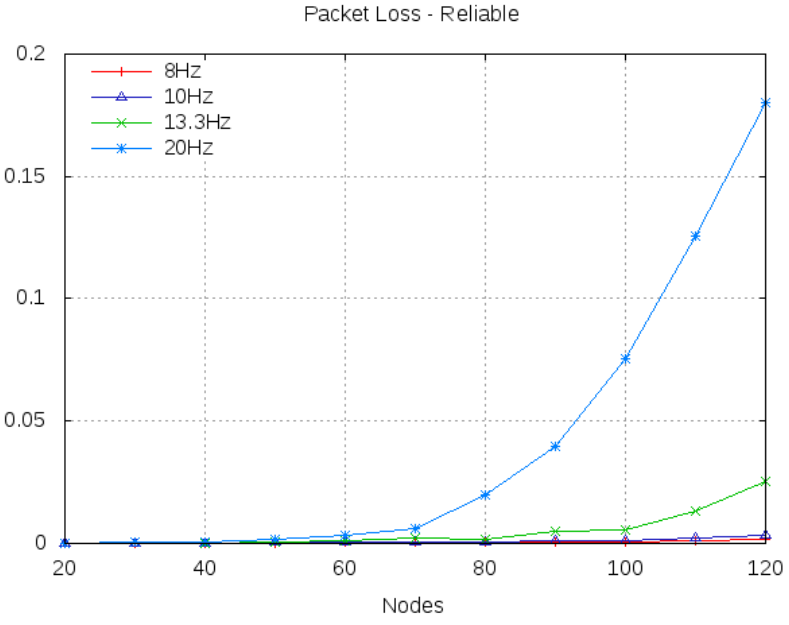

Figure 3.7. Packet Loss using NMP.

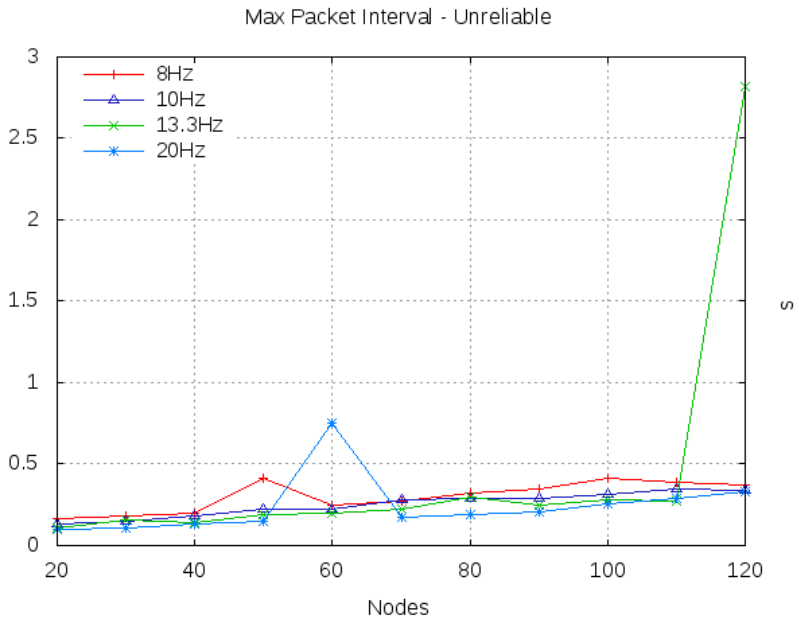

Figure 3.8. Max Packet Interval without NMP.

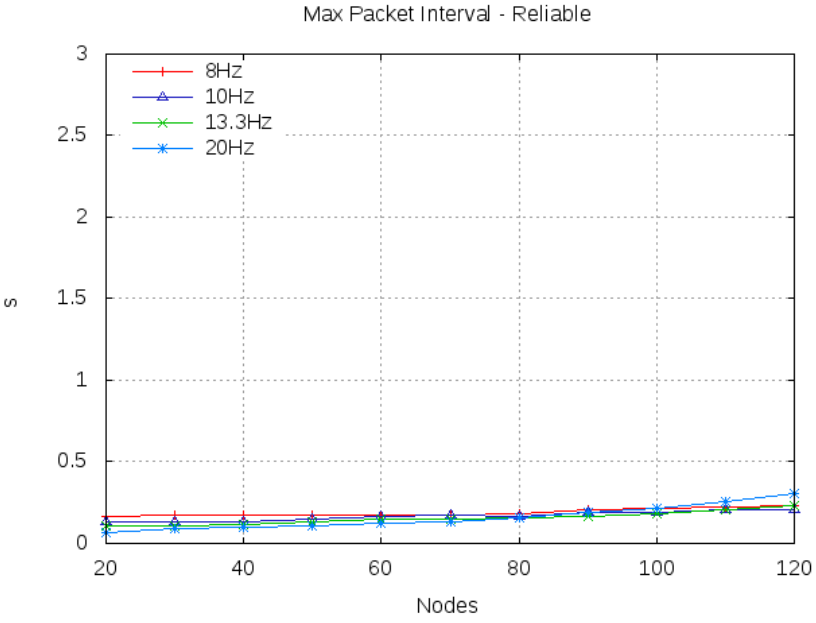

Figure 3.9. Max Packet Interval using NMP.

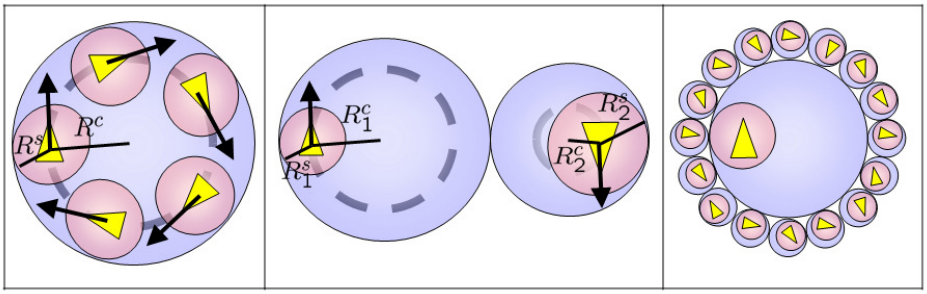

Figure 3.10. R_c and R_s of a mobile device.

The agent has a safety disc of radius R^s centered in the agent itself that must be kept disjoint from safety discs of other agents to avoid collisions.

The GRP policy is based on the concept of *reserved region*, over which each active agent claims exclusive ownership: the circle it would describe under the action of a constant control input $\omega = -\frac{u}{R^c}$, see left of Figure 3.10. In other words, the reserved region for the i-th agent is defined as a disc of radius $R_i^c + R_i^s$ centered at $(x^c, y^c) = (x + R^c \sin(\theta), y - R^c \cos(\theta))$.

$$R_i(t) = \{(x, y) \in \mathbb{R}^2 : \|(x, y) - (x^c, y^c))\|_2 \leq R_i^c + R_i^s\}. \tag{3.4}$$

The motion of the point (x_i^c, y_i^c) is described by the following equations:

$$(\dot{x}_i^c(t),\ \dot{y}_i^c(t)) = (u_i + R_i^c \omega_i(t))(\cos\theta_i(t),\ \sin\theta_i(t)). \tag{3.5}$$

Furthermore, we associate a heading angle to the reserved disc that coincides with the agent heading θ_i. Our policy is based on the following basic observations: the reserved region (i) can be stopped at any time, by setting $\omega = -\frac{u}{R^c}$, see Figure 3.10, and (ii) once stopped, it can be moved in any direction, provided one waits long enough for the heading θ to reach the appropriate value.

A sufficient condition to ensure safety is that the interiors of reserved regions are disjoint at all times; if such a condition is met, conflicts can be avoided if agents hold their reserved regions fixed, and move within them (by setting $\omega = -\frac{u}{R^c}$). As a consequence, each point of contact between reserved regions defines a constraint on further motion for both agents involved. Hence, constraints can be determined if each agent is aware of the configuration of all agents within an *alert distance* $d_a = 2(\hat{R}^s + 2\hat{R}^c)$ where $\hat{R}^c = \max_j R_j^c$ and $\hat{R}^s = \max_j R_j^s$ are respectively the maximum value of R^c and R^s for all agents; see center of Figure 3.10.

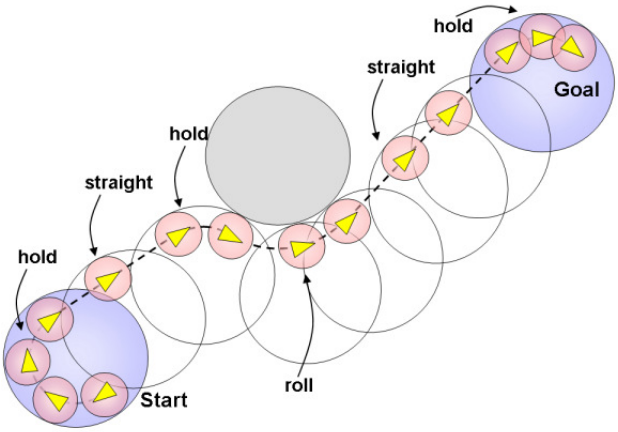

Figure 3.11. Trajectory in case of stationary neighboring reserved region.

For space limitations we report the rule based protocol in Figure 3.12 and who the state changes in presence of an obstacle, Figure 3.11 without other details. The proposed protocol is a modified version with respect to the GRP strategy proposed in [42] in a switching condition between `hold` and `roll`.

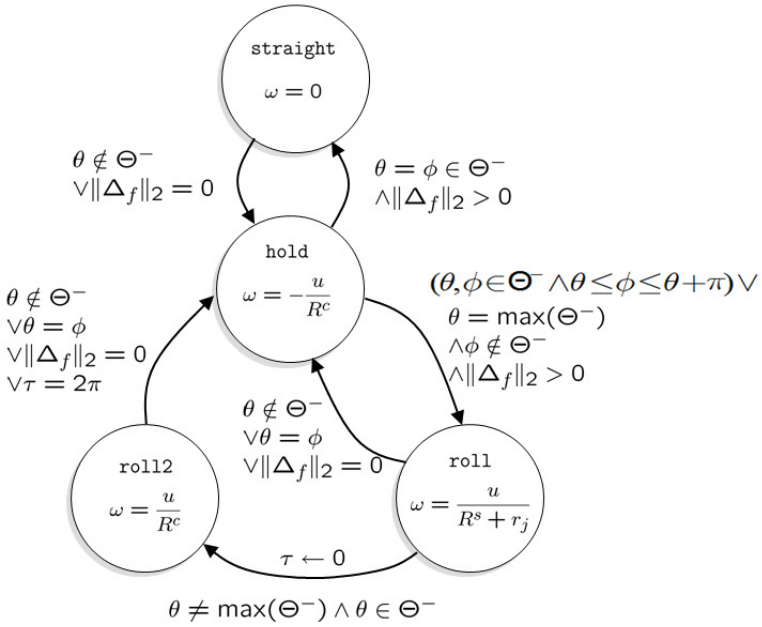

Figure 3.12. A hybrid automaton describing the collision avoidance policy.

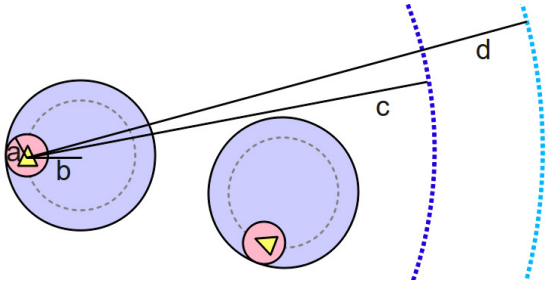

Figure 3.13. a) Safety radius R^s, b) Minimum curvature radius R^c, c) Neighborhood radius D_n, d) Limit distance D_i.

3.5.2 Performance Evaluation of the Case-Study

We have simulated the collision avoidance algorithm and NMP by means of Omnet++ and the INET Framework.

Simulations consider a 10 m $\times$ 10 m shared area where each agent move with a constant linear speed u equal to 5 cm/s. D_n and D_i are respectively fixed to 2 m and 3 m for all agents. Every agent has $R^s = 10cm$ and whereas values of R^c vary between 0 cm (holonomic) and 20 cm. So simulations encompass an heterogeneous set of both holonomic and nonholonomic agents. The communication module emulates a 802.11g NIC in ad-hoc mode with a 2 Mbps transmission rate. The state dissemination frequencies are $F_{max} = 20\ Hz$ and $F_{min} = 0.33\ Hz$.

In order to implement the collision avoidance on NMP we have solved two important implementation issues. The first of them was the *contact condition*. We must initially observe that, theoretically, the collision avoidance algorithm reasons upon the tangency of reserved regions. However, in practice it is impossible. Therefore, we practically assume that two reserved regions are in contact when they are closer than $d_t = 4 \times \tau \times u_M$. This is a conservative value for the worst case.

The second issue was the relationship between the radii, Figure 3.13. The value of the neighborhood radius D_n depends on application features. In this case, in order

Table 3.1. Summary of the notation.

NMP		GRP	
F_{max}	Max. State packet frequency	u	Linear velocity
F_{min}	Min. State packet frequency	ω	Angular velocity
τ	$1/F_{max}$	R^s	Security radius
D_n	Neighbourhood radius	R^c	Curvature radius
D_i	Limit distance		
D_c	Communication radius		

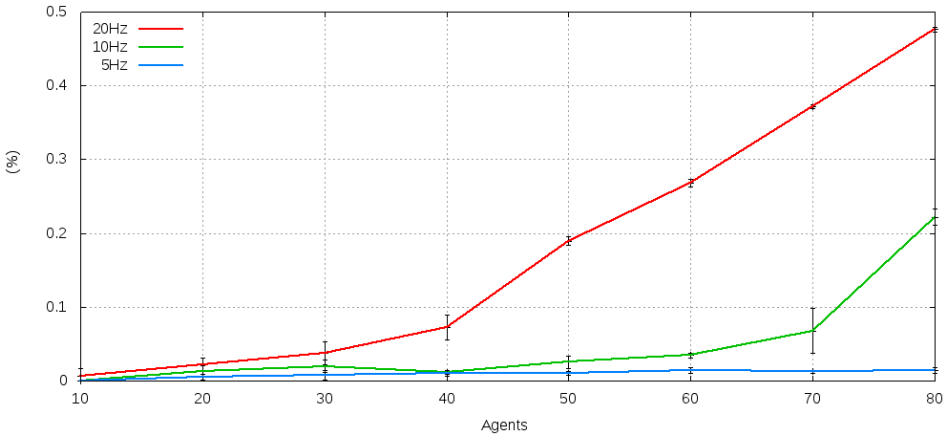

Figure 3.14. Packet loss vs. number of agents. F_{max}: 20Hz, 10Hz and 5Hz.

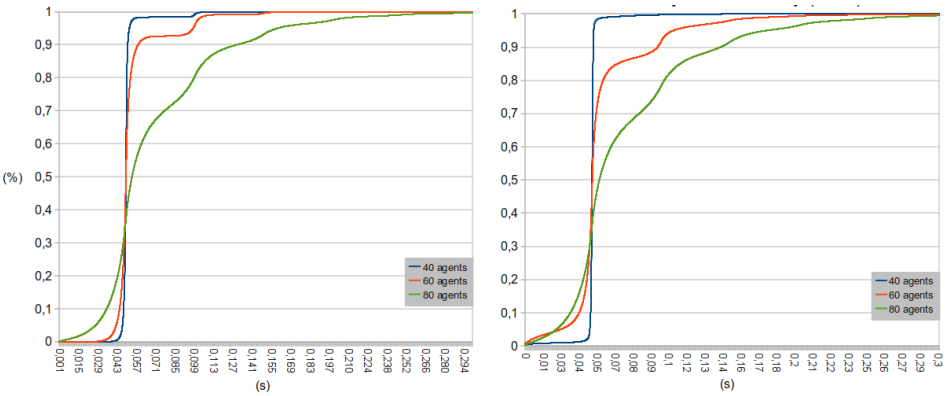

Figure 3.15. Delivery ratio vs. delay. Left: unreliable protocol. Right: NMP.

to preserve the contact condition, a agent's neighborhood must be large enough to encompass the reserved region of a neighbor. It means that $D_n > 4R^c + 3R^s$.

In order to evaluate NMP efficiency and accuracy, we refer to three factors: packet loss, delivery ratio and reserved regions overlapping. The *packet loss* is defined as the ratio between the number of lost packets due to packets collision and the total number of sent packets during an execution of the collision avoidance algorithm. The *delivery ratio* as function of the delay between two consecutive receptions of STATE packet from the same neighbor. It shows the probability that a packet arrives within a specific time interval. Ideally, all packets should arrive with a interval equal to τ. The *reserved regions overlapping* is a sufficient condition for avoiding collisions (see Section 3.5.1) so it tests the system integrity.

Fig. 3.14 shows the packet loss versus the number of agents in the same communication radius. Less than 10% with 40 agents and $F_{max} = 20$ Hz and it increases rapidly with the number of agents.

We have evaluated the NMP behaviour varying the number of agents and we have identify three configurations that reflect the cases of low, medium and high packet loss, respectively with 40, 60 and 80 agents. Usually, the actual collision avoidance applications operate at lower frequencies (2-5Hz). The packet loss is very small if not negligible in the most of practical cases as shown in fig. 3.14 and it allows a greater number of agents managed simultaneously. But, in order to test under stress the communication protocol, we have set our simulations with higher dissemination frequency (20Hz).

Fig. 3.15 shows the delivery ratio using an unreliable broadcast dissemination protocol and using NMP. The delivery ratio gets worse with the number of agents. We have observed that NMP has better performance than the unreliable protocol in case of low packet loss (40 agents), because NMP can easily restores the lost STATE messages if the channel is not congested. On the contrary, in a crowded scenario, NMP contributes to get worse the communication, due to the control messages (NACK packets) that increases the traffic amount. In this case, the unreliable protocol overcomes NMP performance.

Basing on these results, NMP is not recommended in a heavy congested scenario, but it can restore sporadic STATE packets loss that occur in a unreliable dissemination protocol. This small packet loss could lead the system in an inconsistent state. Analyzing the reserved regions overlapping, there are no violation of the safety conditions with NMP in the case of 40 agents. Even if the STATE packet interval is slightly delayed, it is tolerated by collision avoidance algorithm. Otherwise, the unreliable dissemination protocol can experience a single state transmission that is heavy delayed, causing the reserved regions overlapping. In the other two cases with medium and high packet loss, the reserved regions overlapping occurs in both NMP and unreliable broadcast protocol.

4

Localization

The measurement of the distance between two electronic devices is crucial for many practical applications. Many techniques have been proposed over the years [34]. All these techniques fail in the presence of an adversary that wants to disrupt the distance measurement process. Even the well-known and widespread civilian Global Positioning System (GPS) is extremely fragile in adversarial scenarios [30]. Secure location estimation has a plethora of applications including coordination of autonomous guided vehicles as just seen in Chapter 3 and geographical routing [29, 64]. For all these applications, an insecure distance or position estimation could produce security problems such as unauthorized accesses, denial of service, thefts, integrity disruption with possible safety implications and intentional disasters.

Desmedt [14] first introduced the problem of secure location verification and showed that it cannot be solved by solely using cryptography. Brands and Chaum [6] proposed the first *secure distance-bounding* protocol. Since then, many variants have been proposed in the literature [7, 53, 41]. These protocols leverage on both the unforgeability of authenticated messages and the upper bound of the communication speed that is the speed of light. They prevent *distance reduction*, i.e., an adversary cannot make a device appear closer than it really is. The resistance against distance reduction is an important requirement for all the application scenarios involving secure proximity verification [26, 19, 22, 23]. A common example is the problem of proximity-based access control. Let us suppose an RFId card performing an authentication protocol with a reader. If the card correctly performs the protocol, the reader will open a door of a building. An adversary can trick the system by establishing a relay link between the reader and a far away legitimate card, owned by an unaware user. The card correctly performs the authentication protocol via the relay link, and the reader opens the entrance. This attack is known as *mafia fraud*. Along with the correctness of the authentication, the reader has to check even that the card is within a security

distance. However, if such a distance measurement is made with insecure methods, the adversary can still break the system. In particular she can perform a distance reduction attack to deceive the reader into believing that the far away card is in the proximity.

The relevance of the secure proximity verification eclipsed the dual problem: the *distance enlargement* attack. By this attack, an adversary makes a device appear farther than it really is. The resistance against both reduction and enlargement attacks is important whenever we want to securely estimate a distance, rather than a proximity. Let us suppose a distributed system that monitors the movement of autonomous guided vehicles. The system relies on distance information to avoid collisions between vehicles. An example of such systems is the case-study in Section 3.5. If an adversary is able to make a distance appear larger than it really is, the system could not take collision-avoidance countermeasures in time. This could cause collisions between vehicles, and consequent loss of money and safety threats. Secure distance estimations are extremely useful in trilateration techniques too. These techniques use the distances measurements from at least three anchor nodes, whose positions are known, to estimate the position of a fourth node. If an adversary can enlarge one or more distance measurements, she is able to disrupt the whole positioning process.

We propose SecDEv (SECure Distance EValuation), a distance-bounding protocol able to resist to enlargement attacks based on jam-and-replay tactics [31, 60, 59]. SecDEv exploits the characteristics of wireless signals to establish a *security horizon* within which a distance can be correctly evaluated (besides measurement errors) and any adversarial attempt to play a jam-and-replay attack is detected. We also show how SecDEv improves the scalability of secure positioning techniques in terms of number of anchor nodes.

4.1 Related Works

Secure localization has a vast applicability in many technological scenarios, but it has showed to be a nontrivial problem. The silver bullet is yet to be found.

Brands and Chaum [6] proposed distance-bounding protocols, in which a *verifier* node measures the distance of a *prover* node. Distance-bounding protocols do not determine the actual distance, but rather a secure upper bound on it. In this way, the actual distance is assured to be shorter or equal to the measured one, even in presence of an adversary. These protocols were created to assure the physical proximity between two devices, and consequently to contrast *mafia fraud* attack [14].

Hancke and Kuhn [26] fitted distance bounding protocols for RFId tags. Their proposal deals with a variety of practical problems such scarce resources availability, channel noise and untrusted external clock source.

Though extensions for RFId's are possible, we focus on more resourceful devices. We assume the clock source is internal and trusted and the channel noise is corrected by FEC techniques.

Clulow et al. [12] focused on a wide variety of low-level attacks which leverage on packet latencies (e.g. preambles, trailers, etc.) and symbols' modulations. PHY-layer preambles are sent before the cryptographic quantities, in order to permit the receiver to synchronize itself to the sender's clock. The preamble of the response is fixed and does not depend on the content of the challenge. A dishonest prover could thus anticipate the transmission of the response preamble to reduce the measured distance. To deal with this problem, Rasmussen and Čapkun [48] proposed full-duplex distance bounding protocols, in which the challenge and the response are transmitted on separate channels. The prover receives the challenge and meanwhile transmits the response. In this way, a dishonest prover cannot anticipate the transmission of the response, without having to guess the payload. In the present of this chapter, we assume the prover to be honest. This permits us to simplify our reference distance-bounding protocol (cfr. Section 4.2). In particular we use a single channel in a half-duplex fashion.

Flury et al. [22] and, more in depth, Poturalski et al. [46] analyze the PHY-protocol attacks against impulse-radio ultra-wideband ranging protocols (IR-UWB), with particular attention to 802.15.4a [50], which is the *de facto* standard. These studies concentrate only on reduction attacks, and estimate their effectiveness in terms of meters of distance reduction. We instead focus on the opposite problem, distance enlargement, which requires different countermeasures.

Chiang et al. [10] proposed the first technique able to mitigate the enlargement attack in case of dishonest prover. The verifier makes two power measurements of the prover's signal on two collinear antennas. Subsequently, it computes the difference of the two measurements. Given the standard path-loss model, if the difference is low, the signal source will be far away. Otherwise it will be near. The idea is that the adversary cannot modify the way the signal attenuates over the distance, thus the distance estimation is trusted. Obviously such proposal relies on the standard path-loss model, which is poorly reliable. The authors claim that if the path loss exponent varies between 2 and 4, an enlargement of more than twice the measured distance is impossible. In this work, we focus on external adversaries. The problem of distance enlargement in presence of internal ones is challenging as well, but falls outside our present scope.

4.2 Reference Distance-Bounding Protocol

A distance-bounding protocol allows a *verifier* (V) to "measure" the distance of a *prover* (P). In its basic form, a distance-bounding protocol consists in a sequence of single-bit challenge-response rounds [6]. In each round, the verifier sends a challenge bit to the prover that replies immediately with a response bit. The round-trip time enables V to compute an upper-bound of the P distance. Then, the distance is averaged on all rounds. Many variants of distance-bounding protocols have been proposed in the literature [7, 41, 53, 26]. Here, we establish a *reference distance-bounding protocol*, similar to those described in [46] for external adversaries. It involves a *request* message (REQ) from the verifier, an *acknowledgment* message (ACK) from the prover, and a final *signature* message (SGN) from the prover. Such a reference protocol is vulnerable to jam-and-replay attacks, as we will show in Section 4.3, and SecDEv (cfr. Section 4.4) will overcome these vulnerabilities.

The request and the acknowledgement convey, respectively, a and b, which are two independent, random and unpredictable sequences of bits. Note that, differently from the original version of distance-bounding protocol, the request and the acknowledgement are frames, rather than single bits. In fact, it is hard to transmit single bits over an IR-UWB channel. This is due to TLC regulation, which poses strict limits to the transmission power. In 802.15.4a [50], for example, every packet is preceded by a multi-bit synchronization preamble. The signature authenticates the acknowledgement and the request by means of a *shared secret* S. What follows is a formal description of the protocol.

REQ $\mathrm{V} \longrightarrow \mathrm{P} : a$
ACK $\mathrm{P} \longrightarrow \mathrm{V} : b$
SGN $\mathrm{P} \longrightarrow \mathrm{V} : H_S(a, b)$

The quantities a, b and $H_S(\cdot)$ are k-bit long. Therefore, the probability for an adversary to successfully guess one of these quantities is 2^{-k}. Such a probability gets negligible for a sufficiently large value of k, which we call the *security parameter*.

The verifier measures the distance between itself and the prover, by measuring the round-trip time $\hat{T}$ between the request and the acknowledgement messages. With reference to Fig. 4.1, we denote by t_{start} the instant when the transmission of REQ begins, and by t_{end} the instant when the reception of ACK ends. We denote by T_e the time interval from the end of REQ reception, to the beginning of ACK transmission. Since ACK does not depend on REQ, T_e does not include any elaboration time. It includes only the time for the antenna to switch from the receive mode to the transmit mode and the necessary hardware delays. We assume T_e to be small and known. Dedicated hardware can fulfill these requirements. We further denote by T_{pkt} the

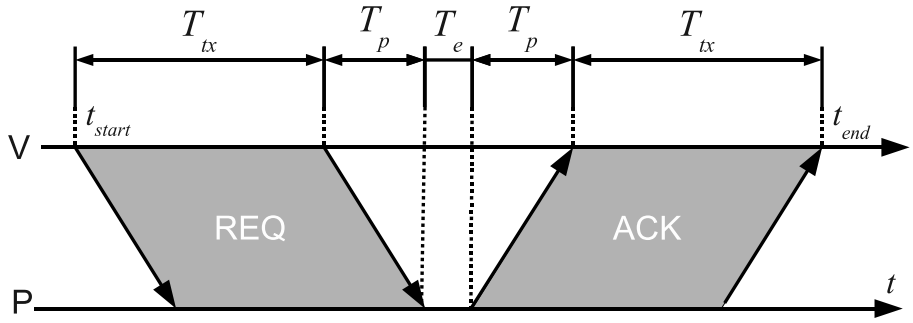

Figure 4.1. Round-trip time.

transmission time of the request and acknowledgement messages, and with T_p their propagation time in the medium. The round-trip time will be:

$$\hat{T} = 2T_p = (t_{end} - t_{start}) - 2T_{pkt} - T_e$$

Finally, we obtain a measure of the distance:

$$\hat{d} = \frac{c \cdot \hat{T}}{2}$$

where c is the speed of light.

The distance measurement precision depends on the capability of measuring the time interval with nanosecond precision. Localization systems based on IR-UWB can achieve nanosecond precision of measured time of flight, and consequently a distance estimation with an uncertainty of 30 cm. Also, this feature of time precision are available only with dedicated hardware.

IR-UWB protocols like 802.15.4a provides packets made up of two parts: a preamble and a payload. The preamble permits the receiver to synchronize to the transmitter and to precisely measure the time of arrival of the packet. The payload carries the information bits. In our protocol, a and b are transmitted in the payload part. We suppose the last part of the payload to carry a forward error correction code (FEC), for example some CRC bits.

In a non-adversarial scenario, the *actual distance* d will be equal to the *measured distance* $\hat{d}$. To deceive the measurement process, the adversary has to bring the verifier to measure a fake round-trip time. That is, she must act in a way that the verifier receives the acknowledgement at a different instant of time, while still receiving the correct signature. The basic idea of distance-bounding protocol is that an external adversary cannot deliver a copy of the legitimate acknowledgement *before* than the legitimate one.

On the other hand, she can deliver a copy of the acknowledgement *after* the legitimate one. In other words, she can only *enlarge* the measured distance, not *reduce* it. Thus, we are always sure that $d \leq \hat{d}$, i.e., the measured distance is a secure upper bound for the actual distance.

4.3 Threat Model

We assume that the adversary (M) is an external agent, meaning that she does not know the shared secret (S) ant it cannot be stolen. Techniques like trusted hardware and remote attestation can help defending against these possibilities [41, 28]. The objective of M is to deceive the verifier into measuring an enlarged round-trip time:

$$\hat{T} = 2T_p + \Delta T \tag{4.1}$$

in order to make it infer an enlarged measured distance:

$$\hat{d} = \frac{c \cdot \hat{T}}{2} = d + \frac{c \cdot \Delta T}{2}$$

We do not deal with distance reduction attacks. Since our protocol is an enhancement of the reference distance-bounding protocol of Section 4.2, it offers the same guarantees against distance reduction attacks.

4.3.1 Adversary's Capabilities

M can eavesdrop, transmit or jam any signal in the wireless channel. The principle of a jammer [63] is to generate a radio noise at a power comparable or higher than the legitimate one. In case of IR-UWB channels, a jammer could send periodic UWB pulses, in such a way to disrupt the synchronization process [45]. Alternatively, she could simply send random pulses in the payload part, in such a way the receiver discards the packet as corrupted after the FEC test. In both cases, the goal of the jammer is to disrupt the reception of the message.

M can transmit or jam *selectively*, in such a way that only a target node receives. In the meanwhile, M can correctly eavesdrop other signals. To do this, she can place a transmitting device nearby the receiver, and a listening one nearby the transmitter. Alternatively, she can use a single device with two directional antennas. One of them transmits to the receiver, while the other listens to the transmitter.

Another possibility is the *overshadowing* attack. In this attack, M injects a fake signal with higher power than the original one. The original signal becomes entirely

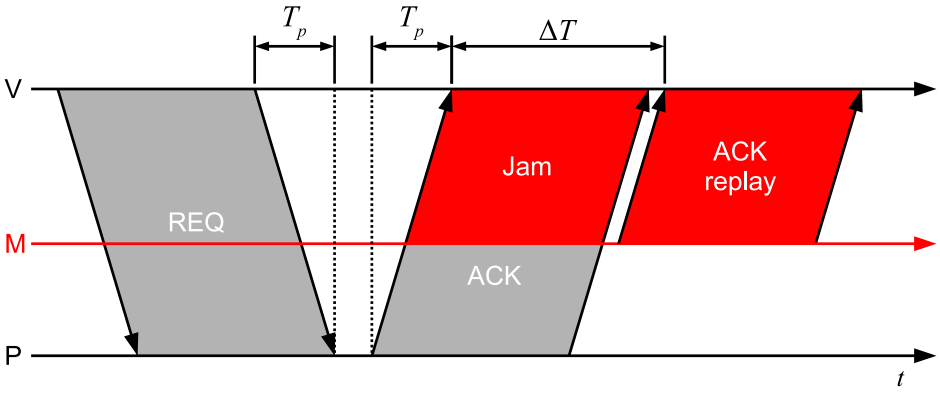

Figure 4.2. Jam-and-replay on ACK.

overshadowed by the attacker's signal. Ideally, original signal is treated as noise by the receiver. In this paper, we do not deal with this attack, and we focus only with jam-and-replay attacks. The overshadowing attack is indeed interesting and deserves a full analysis, that we are planning to do in future work. Here we only points out that it is not simple to be performed in a real-world IR-UWB protocol. In fact, the verifier does not receive only the fake signal, but the legitimate signal too. Even if the former is much stronger in power, the latter is still a valid IR-UWB signal which interferes with the packet synchronization and reception. Sending an overshadowing signal is probably not enough. The adversary should also attenuate the legitimate signal with some complementary technique, such as electro-magnetic shields or similar.

We assume that M has no physical access to the prover or the verifier. This has two consequences: (i) she cannot tamper with the nodes and steal their secret material, and (ii) she cannot attenuate the wireless signals with electro-magnetic shields or Faraday cages.

4.3.2 Jam-and-Replay Attacks

In the distance-bounding protocol of Section 4.2, the adversary can enlarge the measured round-trip time in the following way (Fig. 4.2).

1. M listens to the radio channel, until she hears a REQ signal.
2. M waits for the ACK signal.
3. M jams the ACK signal and eavesdrop it in the meanwhile.
4. After a time ΔT, M replays it.

The adversary must replay the ACK signal selectively, in such a way that only the verifier receives it. Otherwise, the prover will also receive the replayed signal, and could infer that the protocol is under attack.

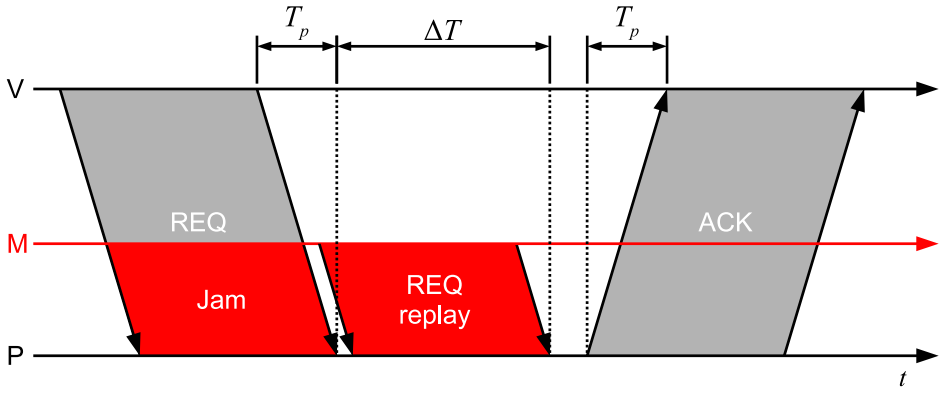

Figure 4.3. Jam-and-replay on REQ.

It is important to highlight that M has to wait for the legitimate ACK to end, before starting the transmission. This is because she must avoid signal collision.

The adversary can perform a similar attack on the REQ signal (Fig. 4.3). Even in this case, M has to wait for the end of the legitimate REQ before starting her transmission.

We state the following:

Proposition 1. *In a jam-and-replay attack on REQ/ACK, the adversary must enlarge the round-trip time of a quantity ΔT not smaller than T_{pkt}, i.e., $\Delta T \geq T_{pkt}$.*

Proposition 1 represents the fundamental limitation of the jam-and-replay attacks. SecDEv will leverage on this to withstand them. Note that this limitation comes from the properties of the radio-frequency channel, and does not depend on how many devices the adversary controls. For the sake of simplicity, Figures 4.2 and 4.3 show a single adversary.

4.4 SecDEv Protocol

SecDEv is a distance-bounding protocol, which measures the correct distance between a verifier V and a prover P in presence of an adversary M performing a jam-and-replay attack. It is similar to the reference distance-bounding protocol (cfr. Section 4.2), except that the length of REQ and ACK do not depend only on the security parameter, but also on a *security horizon*.

Let us consider the Equation 4.1 for a general enlargement attack and apply the Proposition 1, we obtain the constraint $\hat{T} \geq 2T_p + T_{pkt}$. Hence:

Symbol:	Description:
V	Verifier node
P	Prover node
M	Adversary
a	REQ's random bit sequence
b	ACK's random bit sequence
S	Secret material shared between V and P
k	Security parameter
N_{pad}	Number of padding bits in REQ and ACK
N_{fec}	Number of FEC bits in REQ and ACK
$H_S(\cdot)$	Message authentication code of key S
R_{pld}	Transmission bit rate of the payload part
t_{start}	Time instant of REQ's transmission start
t_{end}	Time instant of ACK's reception end
$\hat{T}$	Measured round-trip time between REQ and ACK
T_{pre}	Transmission time of the preamble part of REQ and ACK
T_{pkt}	Total transmission time of REQ and ACK
T_p	Propagation time between V and P
T_e	Response time of V
ΔT	Round-trip time enlargement caused by M
d	Distance between V and P
$\hat{d}$	Measured distance between V and P
d_M	Security horizon
c	Speed of light

Table 4.1. Summary of the notation.

$$\hat{T} \geq T_{pkt} \tag{4.2}$$

Equation 4.2 assures us that a measured round-trip time smaller than T_{pkt} has not been affected by any jam-and-replay attack. We can translate T_{pkt} in a distance d_M, that we call *security horizon*:

$$d_M \triangleq \frac{cT_{pkt}}{2}$$

In terms of distances, Equation 4.2 becomes:

$$\hat{d} \geq d_M \tag{4.3}$$

Equation 4.3 is our test to distinguish between trusted and untrusted distance measurements. V can extend the packet transmission time to enlarge the security horizon (cfr. Eq. 4.3), in order to securely measure longer distances. T_{pkt} is enlarged by introducing padding bits after the nonce, as shown in Figure 4.4. Padding bits

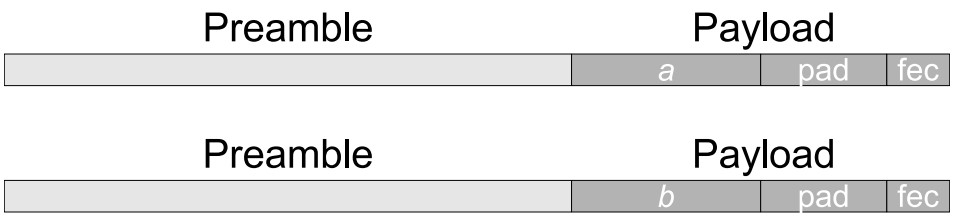

Figure 4.4. SecDEv packet format (REQ and ACK).

have not to be unpredictable. They can have a well-known value (e.g. all zeros), since they serves only to prolong the packet transmission time. V decides on the length of the REQ padding, and P has to respond with the same padding length in the ACK. Therefore, both messages have the same length, to withstand both jam-and-replay on REQ and on ACK.

Let us explain the protocol in detail. We assume that the wireless channel is characterized by the parameter tuple: $\{T_{pre}, R_{pld}, T_e\}$. T_{pre} is the transmission time of the preamble part. R_{pld} is the bit rate of the payload part. T_e is the reaction time of the prover node. In addition, we define the following triplet of protocol parameters: $\{k, S, d_M\}$. k is the security parameter. A higher value for k implies a higher security level, but has an impact on power consumption, as we will see in the following. S is a secret bit sequence shared between V and P. Its length is longer than or equal to k. d_M is the security horizon that distinguishes between trusted and untrusted measured distances. If the actual distance d is longer than d_M, the measured distance cannot be trusted because it may be affected by a jam-and-replay attack. In such a case, the protocol can be executed again with a longer d_M. Alternatively, the distance d can be first estimated in an insecure manner, and then securely confirmed with $d_M > d$. A higher value for d_M allows us to measure longer distances, but has an impact on power consumption.

We further define the following quantities. N_{pad} and N_{fec} are respectively the number of bits of the padding and the FEC code. Since the number of bits of a and b is k, the total transmission time will be:

$$T_{pkt} = T_{pre} + (k + N_{pad} + N_{fec})/R_{pld}$$

If with $N_{pad} = 0$, the T_{pkt} identifies the minimum value of d_M. Thus, if the actual distance is smaller than this value, there is not need of padding bits. Otherwise, we determine N_{pad} with the following formula:

$$N_{pad} = \left\lceil \left(\frac{2d_M}{c} - T_{pre} \right) \cdot R_{pld} \right\rceil - k - N_{fec} \tag{4.4}$$

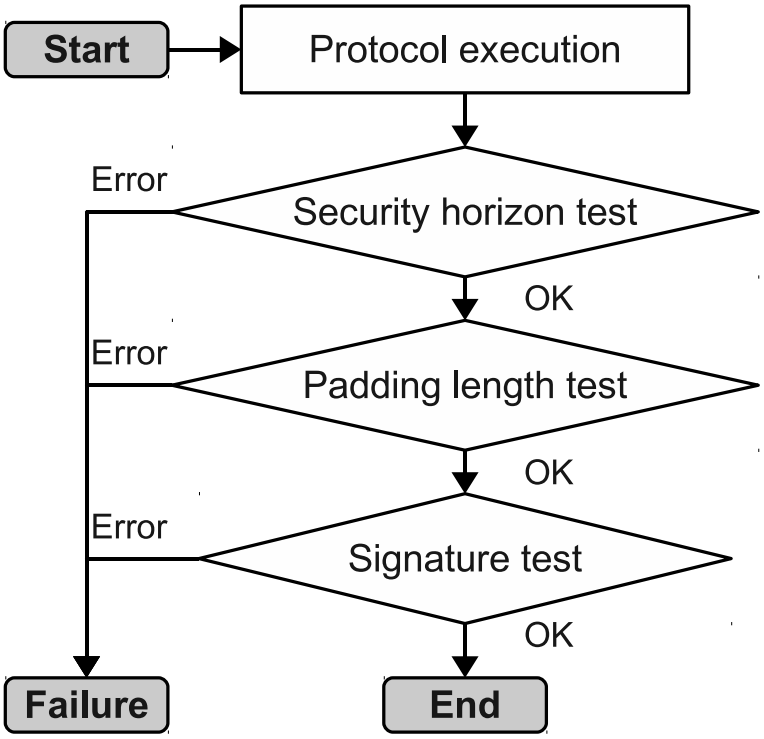

Figure 4.5. SecDEv algorithm.

Using the Equation 4.4, we can set every value of d_M. Note that T_{pkt} grows with d_M. A larger security horizon causes longer messages, accordingly higher energy consumptions per protocol execution. An implementer must choose d_M as a trade-off between ranging capabilities and power consumption.

Figure 4.5 shows the algorithm executed by V. After the protocol execution, V tests whether the measured distance is within the security horizon, that is, if $\hat{d} < d_M$. If this test fails, the measured distance is discarded as untrusted. Then, V tests the length of the ACK padding. If it contains less bits than the REQ one, the measured distance is discarded as untrusted. This is to avoid a jam-and-replay attack on REQ (cfr. Fig. 4.3), in which M tries to lower ΔT by replaying REQ with a smaller padding. In such a case, P will respond with an ACK with a smaller padding too, and the attack will not pass the padding length test. Finally, V tests the validity of the cryptographic signature.

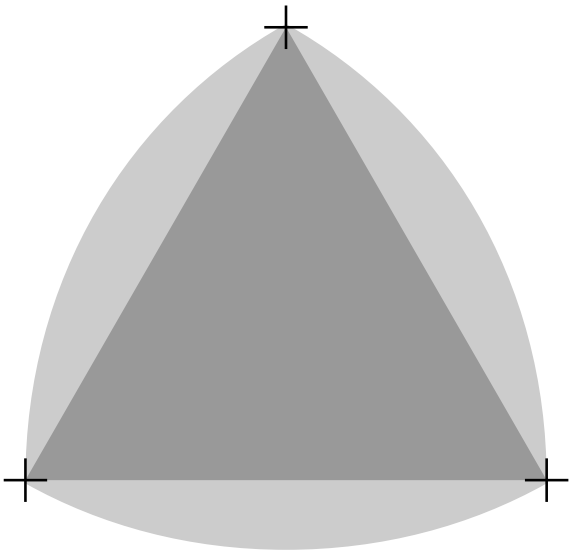

Figure 4.6. Coverage area difference (regular triangle deployment).

4.5 Experimental Results

We combined SecDEv with multilateration technique to securely localize the prover. We analyzed the efficiency of this solution in terms of covered area and we compared it with *verifiable multilateration* [60], which is the state-of-the-art technique for secure positioning in wireless networks. Verifiable multilateration involves at least three distance measurements from different verifiers. The distance measurements are performed by means of distance bounding protocols, which are supposed to withstand reduction attacks. Verifiable multilateration deals with possible enlargement attacks by forcing an additional check to the final position estimation. In order to be trusted, the position must be inside the polygon formed by the verifiers, otherwise it is discarded as untrusted. Intuitively, this reduces the coverage area of the positioning technique.

Figure 4.6 shows the coverage area of verifiable multilateration (in dark grey) and the additional area covered by classic multilateration (in light gray). The verifiers are deployed as a regular triangle's vertices with circular coverage areas and coverage radius equal to the triangle's side, which is the optimal configuration for verifiable multilateration [60]. In Figure 4.6, the coverage improvement of classic multilateration is about 62%. Such an improvement gets even better in Figure 4.7 where the verifiers are deployed randomly.

The following theorem states that, with a generic verifier configuration, the coverage area of classic multilateration is always a superset of the coverage area of verifiable multilateration.

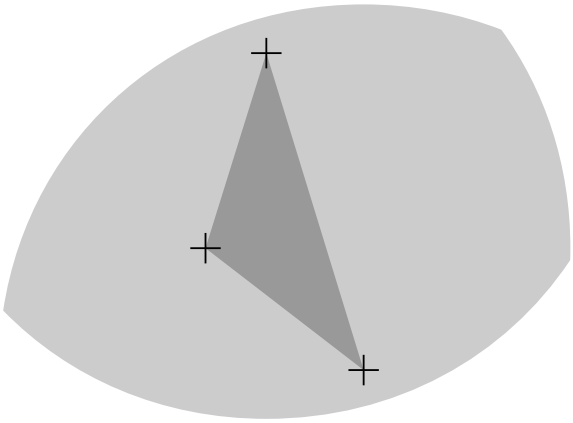

Figure 4.7. Coverage area difference (random deployment).

Theorem 1. *Given a set of distinct positions* $\{X_i : i \leq N\}$*, where* X_i *is the position of the* i*-th verifier, let us call* CM *the coverage area of classic multilateration, and* VM *the coverage area of verifiable multilateration. Then,* $VM \subseteq CM$*.*

Proof. Let us call $R(X_i)$ the coverage area, of the generic verifier X_i. A point is covered by multilateration if and only if it is covered by at least three verifiers. Thus, the total coverage area would be the union of the coverage areas of all the possible triplets of verifiers. Let us consider $\{X_i, X_j, X_k\}$, that is a generic triplet of verifiers. The coverage area of such a triplet with classic multilateration will be:

$$CM_{i,j,k} = R(X_i) \cap R(X_j) \cap R(X_k)$$

Using verifiable multilateration, a prover has to lie inside the verifiers' triangle in order to be correctly localized. Thus, the coverage area of the same triplet with verifiable multilateration will be:

$$VM_{i,j,k} = CM_{i,j,k} \cap T(X_i, X_j, X_k)$$

where $T(X_i, X_j, X_k)$ is the triangular area having the three verifiers as vertices. The total coverage areas with both techniques will be:

$$CM = \bigcup_{i,j,k} CM_{i,j,k}$$
$$VM = \bigcup_{i,j,k} VM_{i,j,k}$$

Since $VM_{i,j,k} \subseteq CM_{i,j,k}$ for each verifier triplet, then:

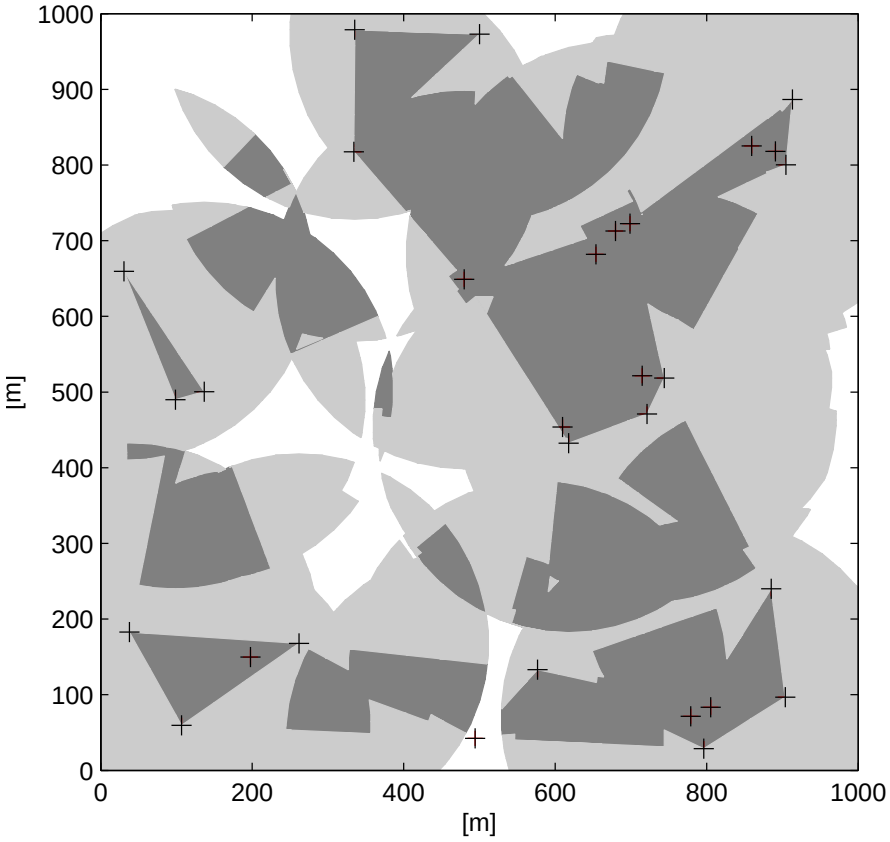

Figure 4.8. Coverage area difference.

$$VM \subseteq CM$$

□

Figure 4.8 shows the coverage area difference with a random distribution of 30 verifiers. The crosses are positions of the verifiers. The dark area is the coverage of verifiable multilateration. The light area is the additional coverage offered by classic multilateration. In other words, classic multilateration is more scalable in terms of number of verifiers needed to cover a specific area. To quantify this, we have tested the performance of classic multilateration in terms of number of verifiers needed to cover a working area, and we have compared our results with those of verifiable mul-

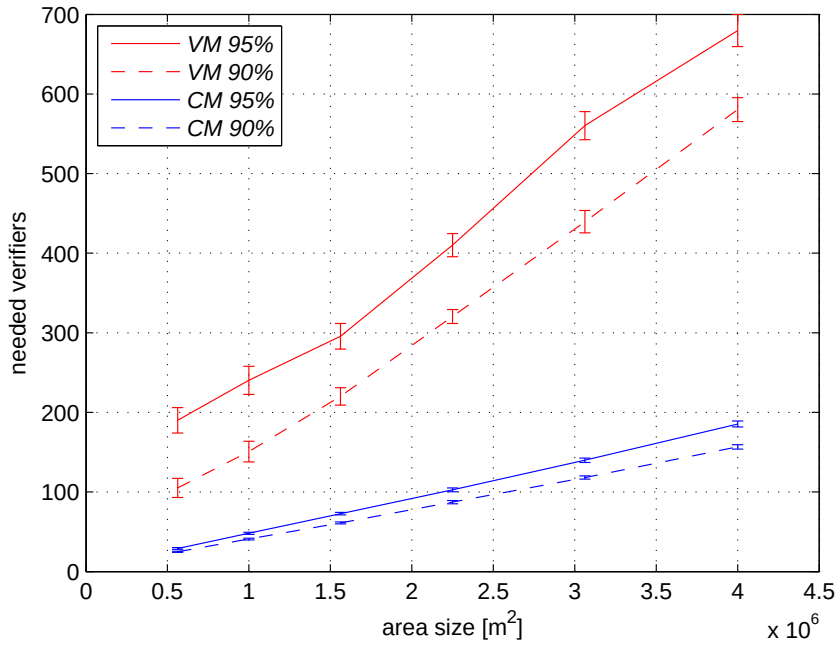

Figure 4.9. Verifiers required to cover an area.

tilateration, taken from [60]. We supposed that every verifier covers a circular area with radius $250\,\mathrm{m}$.

We neglect planned distributions [60], because in a real deployment, environment may impose constraints on the verifier positioning. Thus, we consider that the verifiers are uniformly distributed over the area of interest.

In order to evaluate the two techniques under the same conditions, our simulation were performed on areas of variable sizes. The verifiers were uniformly distributed in the area and in a boundary region outside the area, whose width was 10% of the area width. We use the boundary region to avoid the boundary effects [60] in the verifiable multilateration.

Figure 4.9 shows how many verifiers are required to cover 95% and 90% of the working area. VM and CM curves are respectively verifiable multilateration and classic multilateration. The number of verifiers is the average of 100 simulations with confidence intervals of 95% calculated for different values of working area from $0.5km^2$ to $4km^2$. The chart shows that classic trilateration needs far less verifiers, because it has not the limitation of the verification triangles. This gives strong motivation to fight distance enlargement attacks.

5

Conclusion

In this dissertation, we have faced three security issues which affect the mobile networks: privacy, reliability and secure localization.

We have shown the importance of the privacy in the communication and how it can be achieved by means of the cryptography. As a consequence, it raises the key management issue. Thus, we have presented and analyzed Multi-Group Logical Key Hierarchy, a new centralized and scalable rekeying scheme that guarantees backward and forward security in a multi-group environment, such as a mobile network where subgroups are formed according devices' features and the membership changes dynamically. Our scheme is based on a hybrid approach that combines a logical key tree structure and a logical key star structure. Our simulation showed that MG-LKH improves significantly the communication cost, about one order of magnitude, as compared with the traditional approaches. It means that with the same resources, the system can manage more users or provide more services. Unlike traditional solutions, our scheme has storage, computation and communication requirements that scale in both the group size and number of sub-groups.

The privacy of the communication is not enough if the communication medium is unreliable. We have proposed NMP, a protocol for reliably monitoring the neighborhood of mobile network of AGVs in a industrial system. We have discussed a case-study related to the application of NMP to a challenging collision avoidance algorithm (GRP). Early simulations have shown that NMP is accurate, efficient and scalable due to its efficient use of the wireless medium that takes it to operate very far from the wireless channel saturation zone.

In a mobile network, the localization capability is not trivial and many widespread localization systems have security weaknesses in presence of an adversary. We have proposed SecDEv (SECure Distance EValuation), a distance-bounding protocol able to resist to enlargement attacks based on jam-and-replay tactics. SecDEv exploits

the characteristics of wireless signals to establish a security horizon within which any adversarial attempt to play a jam-and-replay attack is detected. We also showed how SecDEv improves the scalability of secure positioning techniques in terms of number of anchor nodes.

References

1. Control of heterogeneus automation systems: Technologies for scalability, reconfigurability and security. (chat) european commission, 7th framework programme, grant agreement no. fp7-224428, http://www.ict-chat.eu/. Technical report.
2. L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, and L. Seabra Lopes. Coordinating distributed autonomous agents with a real-time database: the cambada project. In *Proceedings of the ISCIS 2004, the 19 th Int. Symp. on Computer and Information Sciences*, 2004.
3. Klaus Becker and Uta Wille. Communication complexity of group key distribution. In *Proceedings of the 5th ACM conference on Computer and communications security*.
4. A. Bicchi, A. Danesi, G. Dini, S. La Porta, L. Pallottino, I.M. Savino, and R. Schiavi. Heterogeneous wireless multirobot system. *Robotics Automation Magazine, IEEE*, 15(1):62 –70, march 2008.
5. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of the Internation Cryptology Conference (CRYPTO)*, pages 258–275, 2005.
6. Stefan Brands and David Chaum. Distance bounding protocols. In *EUROCRYPT'93*, pages 344–359, 1993.
7. Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. *IFIP/SEC*, 2005.
8. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: a taxonomy and some efficient constructions. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 1999.
9. Yi-Ruei Chen, J.D. Tygar, and Wen-Guey Tzeng. Secure group key management using uni-directional proxy re-encryption schemes. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2011.
10. Jerry T. Chiang, Jason J. Haas, Jihyuk Choi, and Yih-chun Hu. Secure location verification using simultaneous multilateration. *IEEE Transactions on Wireless Communications*, 11(2), feb 2012.

11. Kuei-Yi Chou, Yi-Ruei Chen, and Wen-Guey Tzeng. An efficient and secure group key management scheme supporting frequent key updates on pay-tv systems. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, 2011.
12. Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Proceedings of European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS)*, 2006.
13. F. De Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella. On the use of wireless networks at low level of factory automation systems. *Industrial Informatics, IEEE Transactions on*, 2(2):129 – 143, may 2006.
14. Yvo Desmedt. Major security problems with the 'unforgeable' (Feige)-Fiat-Shamir proofs of identity and how to overcome them. *SecuriCom*, 1988.
15. W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 1976.
16. G. Dini and I. M. Savino. S^2rp: a secure and scalable rekeying protocol for wireless sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 457–466, 2006.
17. Gianluca Dini and Francesco Giurlanda. Scalable rekeying in dynamic multi-groups. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 423 –428, june 2010.
18. Gianluca Dini, Francesco Giurlanda, and Lucia Pallottino. Neighbourhood monitoring for decentralised coordination in multi-agent systems: A case-study. *2012 IEEE Symposium on Computers and Communications (ISCC)*, 0:681–683, 2011.
19. Saar Drimer and Steven J. Murdoch. Keep your enemies close: distance bounding against smartcard relay attacks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, 2007.
20. A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi. Decentralized intrusion detection for secure cooperative multi-agent systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 1553–1558. IEEE, 2007.
21. Amos Fiat and Moni Naor. Broadcast encryption. In *Proceedings of the Internation Cryptology Conference (CRYPTO)*, pages 480–491, 1994.
22. Manuel Flury, Marcin Poturalski, Panos Papadimitrios, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Effectiveness of distance-decreasing attacks against impulse radio ranging. In *Proceedings of the third ACM conference on Wireless network security (WiSec2010)*, 2010.
23. Aurélien Francillon, Boris Danev, and Srdjan Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *NDSS*, 2011.
24. IEEE 802.11 Working group. Wireless lan medium access control (mac) and physical layer (phy) specifications, 1999.
25. Qijun Gu, Peng Liu, Wang-Chien Lee, and Chao-Hsien Chu. Ktr: An efficient key management scheme for secure data access control in wireless broadcast services. *IEEE Transactions on Dependable and Secure Computing*, 2009.

26. Gerhard P. Hancke and Markus G. Kuhn. An RFId distance bounding protocol. In IEEE Computer Society Press, editor, *Proceedings of IEEE/Create-Net SecureComm 2005*, 2005.
27. H. Harney and C. Muckenhirn. Group Key Management Protocol (GKMP) Specification.
28. Wen Hu, Hailun Tan, Peter Corke, Wen Chan Shih, and Sanjay Jha. Toward trusted wireless sensor networks. *ACM Transactions on Sensor Networks*, 7(1):1–25, aug 2010.
29. R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *Personal Communications, IEEE*, 8(1):48–57, feb 2001.
30. Roger G. Johnston. Think GPS cargo tracking = high security? think again. Technical report, Los Alamos National Laboratory, 2003.
31. Jiejun Kong, Zhengrong Ji, Weichao Wang, Mario Gerla, Rajive Bagrodia, and Bharat Bhargava. Low-cost attacks against packet delivery, localization and time synchronization services in under-water sensor networks. In *Proceedings of the 4th ACM workshop on Wireless security*, WiSe '05, pages 87–96, New York, NY, USA, 2005. ACM.
32. Ki-Ho Lee and Dong-Ho Cho. A multiple access collision avoidance protocol for multicast service in mobile ad hoc networks. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 3, pages 1793 – 1797 vol.3, april 2003.
33. Xiaozhou Steve Li, Yang Richard Yang, Mohamed G. Gouda, and Simon S. Lam. Batch rekeying for secure group communications. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 525–534, 2001.
34. Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067 –1080, nov. 2007.
35. Yi-Ru Liu and Wen-Guey Tzeng.
36. D. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. RFC 1305 (Draft Standard), mar 1992.
37. D. Mills. Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. RFC 4330 (Informational), jan 2006.
38. M.J. Moyer, J.R. Rao, and P. Rohatgi. A survey of security issues in multicast communications. *Network, IEEE*, 1999.
39. Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of the Internation Cryptology Conference (CRYPTO)*, pages 41–62, 2001.
40. W.H.D. Ng, M. Howarth, Z. Sun, and H. Cruickshank. Dynamic balanced key tree management for secure multicast communications. *Computers, IEEE Transactions on*, 2007.
41. Ventzislav Nikov and Marc Vauclair. Yet another secure distance-bounding protocol. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT'08)*, pages 218–221, 2008.
42. L. Pallottino, V.G. Scordio, A. Bicchi, and E. Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *Robotics, IEEE Transactions on*, 23(6):1170 –1183, dec. 2007.

43. Adrian Perrig. Efficient collaborative key management protocols for secure autonomous group communication. In *In International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pages 192–202, 1999.
44. Adrian Perrig, Dawn Song, and J. D. Tygar. Elk, a new protocol for efficient large-group key distribution. In *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, pages 247–262, 2001.
45. Marcin Poturalski, Manuel Flury, Panos Papadimitrios, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. The cicada attack: degradation and denial of service in ir ranging. In *Proceedings of 2010 IEEE International Conference on Ultra-Wideband (ICUWB2010)*, 2010.
46. Marcin Poturalski, Manuel Flury, Panos Papadimitrios, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Distance bounding with IEEE 802.15.4a: Attacks and countermeasures. *IEEE Transactions on Wireless Communications*, 2011.
47. Sandro Rafaeli and David Hutchison. A survey of key management for secure group communication. *ACM Computing Surveys*, 35(3):309–329, 2003.
48. Kasper Bonne Rasmussen and Srdjan Čapkun. Location privacy of distance bounding protocols. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 149–160. ACM, 2008.
49. A. Fagiolini S. Manca, L. Pallottino and A. Bicchi. Decentralized coordination system for multiple agvs in an industrial environment based on shared resources. In *In IEEE Conference on Automation Science and Engineering (CASE 2010)*.
50. Zafer Sahinoglu and Sinan Gezici. Ranging in the IEEE 802.15.4a standard. In *Proceedings of IEEE Wireless and Microwave Technology Conference*, 2006.
51. Alan T. Sherman and David A. McGrew. Key establishment in large dynamic groups using one-way function trees. *IEEE Transactions on Software Engineering*, 2003.
52. Shiann-Tsong Sheu, Yihjia Tsai, and Jenhui Chen. A highly reliable broadcast scheme for ieee 802.11 multi-hop ad hoc networks. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 1, pages 610 –615, 2002.
53. D. Singelee and B. Preneel. Key establishment using secure distance bounding protocols. In *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, 2007.
54. Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-hellman key distribution extended to group communication. In *Proceedings of the ACM Conference on Computer and Communications Security*, 1996.
55. Min-Te Sun, Lifei Huang, A. Arora, and Ten-Hwang Lai. Reliable mac layer multicast in ieee 802.11 wireless networks. In *Parallel Processing, 2002. Proceedings. International Conference on*, pages 527 –536, aug. 2002.
56. Yan Sun and K.J.R. Liu. Hierarchical group access control for secure multicast communications. *Networking, IEEE/ACM Transactions on*, 2007.
57. K. Tang and M. Gerla. Random access mac for efficient broadcast support in ad hoc networks. In *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*, pages 454 –459 vol.1, 2000.

58. K. Tang and M. Gerla. Mac reliable broadcast in ad hoc networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, pages 1008 – 1013 vol.2, 2001.
59. Nils Tippenhauer and Srdjan Čapkun. ID-based secure distance bounding and localization. In Michael Backes and Peng Ning, editors, *Computer Security – ESORICS 2009*, volume 5789 of *Lecture Notes in Computer Science*, pages 621–636. Springer Berlin / Heidelberg, 2009.
60. S. Čapkun and J.-P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, feb 2006.
61. D. Wallner, E. Harder, and R. Agee. Key Management for Multicast: Issues and Architectures. RFC 2627 (Informational), June 1999.
62. Chung Kei Wong, M. Gouda, and S.S. Lam. Secure group communications using key graphs. *Networking, IEEE/ACM Transactions on*, 2000.
63. Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '05*, 2005.
64. Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report, UCLA Computer Science Department Technical Report, 2001.