Ministry of higher education and scientific research
University of Abou Bekr Belkaid
Faculty of engineer
Department of electronics

Master Thesis

Presented by

KADRI Benamar

The adaptation of security mechanisms for Ad hoc Networks

Thesis director Prof. Feham.M

Members of Jury:

Mr. Benahmed N.E.	University of Abou Bekr Belkaid, Tlemcen, Algeria. President.
Mr Chikh AZ.	University of Abou Bekr Belkaid, Tlemcen, Algeria. Examiner.
Mr abdelmalek A.	University of Abou Bekr Belkaid, Tlemcen, Algeria. Examiner.
Mr Abdellah M.	National Institute of Telecommunications, Evry, France, Co-Thesis director.

Dedication

My first and great dedications are for the most precious thing in my life of course my parents. I hope for them long life to assist to the realization of our dreams.

I would like also to dedicate this work to my brothers Mustapha and Abderazak the best company that I ever had, to my sister Siham who has got her baccalaureate I hope for her good luck in the university.

To all those who love me

I tell them I love you all.

Benamar.

Acknowledgements

First and foremost I thank Allah, because without the help of Allah this work could never be materialized.

I would also to thank my two supervisors, Mr M'hamed Abdellah from the national institute of telecommunication in France for his guidance and infinite patience, for the encouragements given for me along of this work; I would also thank him for the precious help that he has given me for my papers and my thesis.

My dept thanks to Mr Feham Mohamed from the STIC laboratory who has given me the necessary encouragements and orientations to achieve this modest work, I want also to appreciate his characters to encourage his student for working, by giving them the ability to create and invent to elaborate the best work.

I would also to thank the members of jury (Mr Benahmed, Mr abdelmalek, Mr chikh) to accept judging this work and for their patient when reading it.

Lastly I wish to thank the members of STIC laboratory each one by his name, for their friendship which gives us more ability to work and for their help and encouragement.

Abstract

Mobile Ad Hoc networks (MANETs) gain day after day places in our life considering the benefits given by these networks as mobility, facility of deployment and their costless. However their benefits are also subject of new challenges regarding the characteristics of these networks as battery and bandwidth constraints…etc.

Therefore, new MANETs must confront lot of problems as security, routing, quality of service and energy saving. However from our point of view we see that the problem of security is the most important problem since mobile nodes are often part of hostile environment exposing them to new risks and attacks making private life or confidential information (commercial or military) in danger.

Regarding the characteristics of mobile nodes as battery and processor power, existed and effective solutions developed for conventional networks are inapplicable directly for MANETs. Therefore in this thesis we try to give an adaptation of one of the most effective and efficient tool providing security for conventional networks which is Public Key Infrastructure (PKI), since it still questionable if it can be implemented for MANETs or not.

Our proposed solution to adapt PKI for MANETs uses clustering, since recent researches have proven that clustering is effective to manage the complexity and the increasing size of MANETs, by dividing the whole network on a set of clusters (groups) in which one of the members plays the role of cluster-head managing its cluster and inter cluster communication. In our solution the cluster-head is the Certificate Authority of its cluster, guarantying in this way a permanent connection between the Certificate authority and the users. We've also developed and tested a new clustering algorithm called Secured Clustering Algorithm in order to support the security needs of clustered PKI, by including a set of security parameters in the creation of clusters.

Key words: MANET, Security, Key Management, PKI, Clustering.

Table of contents

Chapter II State of the art

Chapter III Cluster based PKI

Chapter IV Secured Clustering Algorithm

List of Figures

List of graphs and tables

General Introduction

Wireless networks are a collection of wireless mobile hosts forming a temporary network, using as transmission medium radio waves. The used spectre for wireless transmissions is the spectre situated around the 2.4 GHz ISM (Industrial, Scientific and Medical), and around the 5 GHz U-NII (Unlicensed-National Information Infrastructure). The transmission range and the emission power are regulated by laws in each country depending on the location where the network is deployed (indoor or outdoor), ranging from 10 m for Personal Area Networks to 100-200 m for Local Area Networks.

Regarding its topology different configurations can be defined for wireless networks, in our work we are interesting on Mobile ad hoc network (MANET). MANETs are a collection of wireless hosts that communicate with each other through multi-hop wireless links, without the existence of any infrastructure or administrative authority. Therefore nodes must collaborate between them to accomplish some operations like routing and security.

Regarding its costless, facility of use and deployment, MANET gets day after day new applications ranging from military applications for connecting soldier in battlefields and civil or commercial application such as Public and Personal Area Networks, other applications are recently under development will also benefit from MANETs advantages such as telemedicine, weather report and disaster environment such as in seism. All these examples of use predict for some envisioned MANETs to increase in size to reach the threshold of thousands of nodes per system (commercial or military).

Unlike wired networks using dedicated nodes to support basic functions like routing, security, and network management, in ad hoc networks those functions are carried out by all or a subset of available nodes, making these tasks more difficult to carry in MANETs. Therefore any existed mechanism developed for wired networks must be adapted to be used in MANET taking into account the no existence of infrastructure, bandwidth constraint and other aspects due to the nature of mobile nodes (Mobility, power constraint, risk of loss and theft). Regarding their characteristics new areas of researches have appear to treat some specific problematic for MANETs:

Routing, this area of research has to adapt or develop effective protocols for MANETs. The researches on this domain have given birth to new protocols assumed to be effective only for some specific configuration of ad hoc networks. Therefore the domain of routing still in development and new protocols appear each day until an effective and efficient protocol taking all MANETs characteristic will be developed.

Quality of Service, as it is a great problematic for wired networks; Quality of Service has more challenges to confront in MANETs such as topology changing due to mobility, making

conventional protocols practically inapplicable for MANETs. Therefore the development of new innovative protocols is unavoidable.

Security, in contrary to wired networks, MANETs nodes are more often part of a hostile environment that is not maintained professionally exposing the network to new risks ranging from physical attacks (theft) to eavesdropping due to the transmission range which often exceeds the area where the network is deployed.

Energy, this area of research focuses the adaptation of all protocols (Routing, Quality of Service, Security…etc) to be energy award. In order to regulate the use of battery to avoid quick battery drain.

As in wired network the primordial need in MANETs is the need of security, since we can't benefit from any effective routing or Quality of Service protocol if the MANETs aren't well protected making private or confidential information exchanged over the network available to attackers.

Therefore in this modest work we try to study and find solutions to adapt the Public Key Infrastructure (PKI) for MANETs. PKI has been recognised as the most efficient and effective tool providing key management in classical networks guarantying authentication, privacy, no repudiation and integrity using asymmetric encryption. However providing such infrastructure for ad hoc network is a challenging task, due to the dynamic topology and infrastructure-less nature of these networks. In this thesis we propose a new scheme for the implementation of PKI infrastructure in ad hoc network based on clustering technique in order to simplify the key management, since clustering has been supposed to be effective for some routing protocols taking advantages of grouping the network into small groups to facilitate the network management. Thus we think that using clustering to manage security may also benefit from these advantages. Since the existed clustering algorithm aren't developed to serve security purposes we've also developed a new clustering algorithm called Secured Clustering Algorithm, intended to serve security purposes, used as underlying clustering protocol to serve our proposed scheme of PKI infrastructure.

The manuscript of this thesis is organized in four chapters:

In the first chapter, we try to give an introduction to ad hoc networks in order to fix the target of our work. This chapter is organized into two parts, the first part is a brief definition of all ad hoc networks configurations, characteristics and their area of use and application, and then we exhibit our interesting and the primordial problematic of these networks which is security. In the second part we expose a cryptography background which is necessary to develop any mechanism of security for any system. By exposing different algorithms and cryptography techniques such as symmetric and asymmetric as well as some security mechanism developed for wired networks such as PKI.

In chapter two, we give an overview of existed solutions to ensure security in ad hoc networks. In the way that we give a set of key management schemes, then we analyse each one by giving its advantages, disadvantages and its possibility of use for ad hoc networks. This outgoing of existed work of key management schemes for ad hoc network is divided on two parts in the first part we present the PKI based management scheme like partially and fully distributed certificate. However in the second part we give other mechanisms based on other techniques like location limited channels and symmetric encryption.

In chapter three, we try to expose our contribution to ensure security in ad hoc networks. As we've said our solution for key management in ad hoc networks is a PKI management scheme based on clustering. Thus chapter III begins by situating the problem by giving network models and assumptions on network configuration and its mobile nodes capabilities. Then we expose our protocol design by defining all its aspects as system bootstrapping, cluster creation, key and certificate generation; revocation or renewal…etc. Then we give a brief analysis of our protocol against some known attacks, its respect of the key management characteristics and a comparison with existed key management schemes cited in chapter II. In the last part of this chapter we expose the prototype implementation which is done in JAVA to give for our prototype more scale of area for tests. This prototype is mainly developed to observe the reaction of our protocol to some situations in ad hoc networks.

In chapter four, we expose our proposed clustering algorithm intended to serve security protocols such as key management. We begin this chapter by giving a brief outgoing of recent and known works in the domain of clustering, following each algorithm by an analysis in which we describe its limitations and disadvantages. Then we expose our proposed algorithm Secured Clustering Algorithm, in which we've tried to include the aspect of security by proposing a voting mechanism, intended to define how much any node is trusted by its neighbourhood to be elected or not as cluster-head. Other system parameters are included in our algorithm making its efficient and adaptable for different configurations of ad hoc networks. In the last section we try to compare using ClusterSIM our proposed algorithm with existed algorithm regarding the number of clusters created and the number of nodes in each cluster, we also try to test the performance of our protocol for different size and configuration of ad hoc network regarding the size and the number of nodes.

Chapter I

Introduction to Ad Hoc networks and cryptography

Ad hoc networks know day after day lot of area of use; otherwise they have lot of challenges to defy as security, routing and management. Therefore in this first chapter we try to briefly present ad hoc networks and their problematic. This chapter is divided into two parts.

The first part gives a brief introduction to ad hoc network in which we are going to present different kinds and configurations of these networks; then we'll present the most known applications using these networks. The last section of this part gives an introduction to the most known problematic of ad hoc networks which are routing and security.

The second part of this chapter gives a cryptography background giving the reader of this thesis the basis to continue the rest of chapters. In the first sections of this part we are going to present different techniques used in cryptography as symmetric and asymmetric encryption. We consecrate the last sections for the presentation of Public Key Infrastructure and its important components; to finish this chapter by posing the problem of using this infrastructure for ad hoc networks.

I- Presentation of wireless network

I-1 Introduction

A wireless network is a set of computers or any other devices like PDA, cellular phones…etc, using as transmission medium the radio waves. The used spectre for wireless transmissions is the spectre situated around the 2.4 GHz ISM (Industrial, Scientific and Medical) band for a bandwidth of about 83 MHz, and around the 5 GHz U-NII (Unlicensed-National Information Infrastructure) band for a bandwidth of about 300 MHz. The exact frequency allocations are set by laws in the different countries; the same laws regulate also the maximum allowed transmission power in deferent location (indoor, outdoor) [1].

Such a wireless radio network has a range of about 10–100 meters to 10 Km per machine, depending on the emission power, the data rate, the frequency, and the type of antenna used. Many different models of antenna can be employed: omni (omni directional antennas), or sector antennas (directional antennas) [2].

The other type of transmission support is the infrared. Infrared rays cannot penetrate opaque materials and have a smaller range of about 10 meters. For these reasons, infrared technologies are mostly used for small devices in WPANs (Wireless Personal Area Networks), for instance to connect a PDA to a laptop inside a room [1].

I-2- Standards

The IEEE (Institute of Electrical and Electronics Engineers) initiated the 802.11 project in 1990 with a scope to develop a Medium Access Control (MAC) and Physical Layer (PHY) specifications for wireless networks. In 1997, IEEE first approved the 802.11 international interoperability standards. Then, in 1999, the IEEE ratified the 802.11a and the 802.11b wireless networking communication standards.

Although the rest of this work we focus on the IEEE 802.11 WLAN (Wireless Local Area Networks) standard, it is important to note that several other WLAN technologies and standards are available from which consumers may choose, including HiperLAN and HomeRF.

There are presently four main standards for wireless networks [2]: the IEEE 802.11, IEEE 802.15, IEEE 802.16, and IEEE 802.20.

I-2-1 IEEE 802.11

a- Presentation

IEEE 802.11 is a standard issued by the IEEE. It is designed for WLANs. From the point of view of the physical layer, it defines three non-interoperable techniques: IEEE 802.11 FHSS (Frequency Hopping Spread Spectrum) and IEEE 802.11 DSSS (Direct Sequence Spread Spectrum), which use both the radio medium at 2.4 GHz, and IEEE 802.11 IR (InfraRed). It has also modified the link layer by changing the mechanism of accessing the physical layer since the physical layer has changed and keep the LLC (*Logical Link Control*), the same as in 802.2 (Ethernet) which gives it the possibility to operate with Ethernet network which are widely deployed. This specification has given birth to a family of other standards:

-IEEE 802.11a: (marketed as Wi-Fi5) operates in the 5 GHz U-NII band using the OFDM (Orthogonal Frequency Division Multiplexing) transmission technique, and has a maximum data rate of 54 Mbps. IEEE 802.11a is incompatible with 802.11b, because they use different frequencies.

-IEEE 802.11b: (marketed as Wi-Fi) is the de facto standard in wireless networking, and operates in the 2.4 GHz ISM band. The data rate is 1, 2, 5 or 11 Mbps, automatically adjusted

depending on signal strength. The transmission range depends on the data rate, varying from 50 meters indoor (200 meters outdoor) for 11 Mbps, to 150 meters indoor (500 meters outdoor) for 1 Mbps; the transmission range is also proportional to the signal power.
-IEEE 802.11g: operates in the 2.4 GHz band and has a data rate of up to 20 Mbps. It uses both OFDM and DSSS to ensure compatibility with the IEEE 802.11b standard.
-IEEE 802.11e: This standard is supplementary to the MAC layer to provide QOS support for LAN applications. It will apply to 802.11 physical standards a, b, and g. The purpose is to provide classes of service with managed levels of QOS for data, voice, and video applications.
-IEEE 802.11f: This standard aims to achieve radio access point interoperability within a multivendor WLAN network. This standard defines the registration of access points within a network and the interchange of information between access points when a user is handed over from one access point to another.
-IEEE 802.11h: This standard is developed to better mange the consummation of power according to the node location (indoor or outdoor). It tries also to manage the legislation of every country.
-IEEE 802.1X and IEEE 802.11i: these two standards are developed to ensure security in 802.11 networks. The 802.1X is a port-level access control protocol, firstly developed for wired networks [2], and adapted to wireless networks. The 802.11i standard, also still in draft, was created for wireless-specific security functions that operate with IEEE 802.1x.

b- Channel access techniques

The crucial point in channel access techniques for wireless networks is that it is not possible to transmit and to sense the carrier for packet collisions at the same time. Therefore there is no way to implement a CSMA/CD (Carrier Sense Multiple Access / Collision Detection) protocol such as in the wired network. IEEE 802.11 uses a channel access technique of type CSMA/CA, which is designed to perform Collision Avoidance (or at least to try to do). The CSMA/CA protocol states that a node, upon sensing that the channel is busy, must wait for an interframe spacing before attempting to transmit, then choose a random delay depending on the Contention Window.

The reception of a packet is acknowledged by the receiver to the sender. If the sender does not receive the acknowledgement packet, it waits for a delay according to the binary exponential backoff algorithm, which states that the Contention Window size is doubled at each failed try.

Unicast data packets are sent using a more reliable mechanism. The source transmits a RTS (Request To Send) packet for the destination, which replies with a CTS (Clear To Send) packet upon reception. If the source correctly receives the CTS, it sends the data packet.

I-2-2 IEEE 802.15

IEEE 802.15 also known as Bluetooth is a standard designed by a consortium of private companies such as Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia and Toshiba for WPANs (Wireless Personal Area Networks). Bluetooth operates in the 2.4 GHz band using FHSS and has a short range of action of about 10 meters. For such characteristics and its low cost, Bluetooth is fit for small WPANs and is also employed to connect peripherals such as keyboards, printers, or mobile phone headsets. Bluetooth radio technology works in a master-slave fashion, and each device can operate as master or as slave. Communications are organized in small networks called *piconets*, each *Piconet* being composed of a master and 1–7 active slaves. Multiple *piconets* can overlap to form a *scatternet*. In the way that a node play the rode of router between the two *piconets*, and is responsible of forwarding data between the

two *piconets*. This intermediate node may be slave or a master in one of the two attached *piconets*.

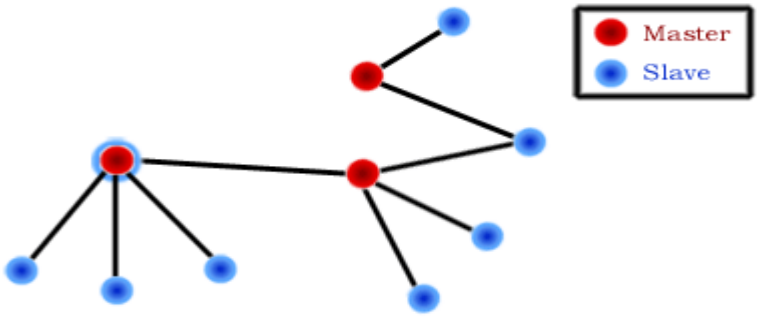


Figure I.1 PAN using Bluetooth technology

I-2-3 IEEE 802.16

This standard is currently under tests [2] (marketed as WiMAX), is designed for WMANs (Wireless Metropolitan Area Networks) and therefore to overcome the range limitations of IEEE 802.11. It operates on frequencies from 10 to 66 GHz, and should ensure network coverage for several square Km.

I-2-4 IEEE 802.20

Also known as Wi-Mobile it operates at the same level as UMTS or CDMA2000. It is designed for WWANs (Wireless Wide Area Networks), it is considered as a concurrent of these two techniques. It may be largely used since it has low cost compared to UMTS, because it use existed techniques already developed like IP and Ethernet [2].

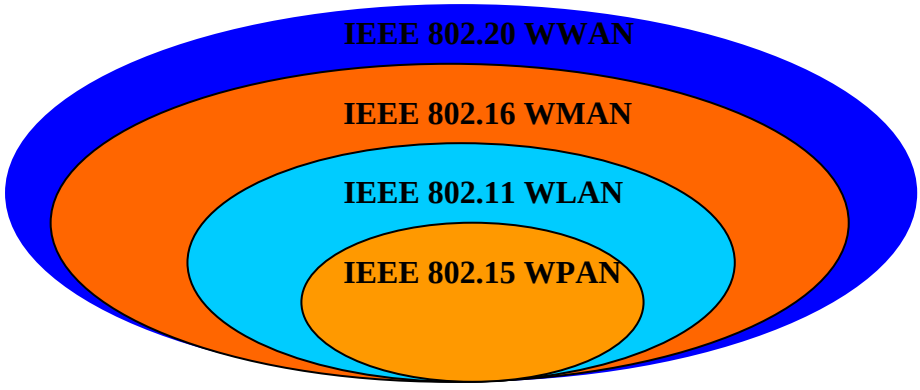


Figure I.2 Wireless standard

I-3- Architectures of WLAN

A wireless network can be structured to function in either BSS (Basic Service Set) or IBSS (Independent Basic Service Set) mode. The two modes affect the topology and the mobility capabilities of the machines (*nodes*) that compose the network.

I-3-1 BSS mode

In BSS mode, also called *infrastructure mode*, a number of mobile nodes are wirelessly connected to a non-mobile Access Point (AP), as in Figure I.3. Nodes communicate via the AP, which may also provide connectivity with an external wired network e.g. the Internet. Several BSS networks may be joined to form an ESS (Extended Service Set).

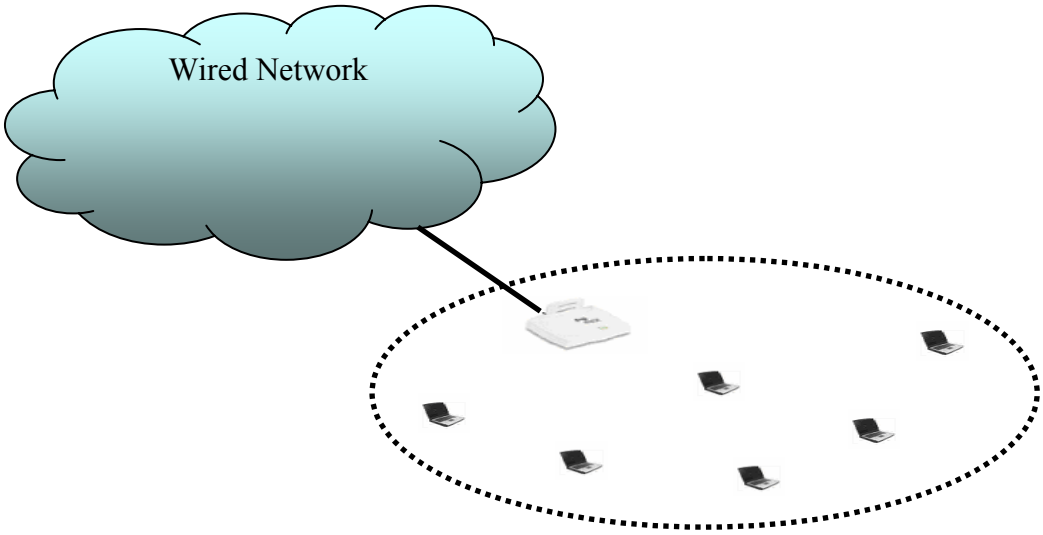


Figure I.3 Infrastructure mode

I-3-2 IBSS mode

The IBSS mode, also called *peer to peer* or *ad hoc mode*, allows nodes to communicate directly (point-to-point) without the need for an AP, as in Figure I.4. There is no fixed infrastructure. Therefore nodes need to be in range of each other in order to communicate.

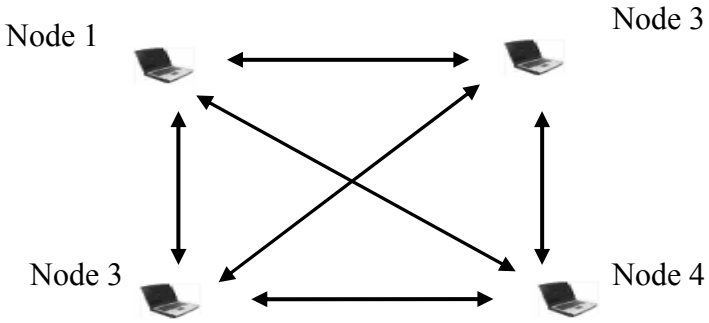


Figure I.4 Ad Hoc Mode

I-4- Ad hoc network

An ad hoc network or MANET (Mobile Ad hoc NETwork), is a network composed only of nodes (PDA, Phones, laptops…), with no Access Point [3, 4, 5]. Messages are exchanged and relayed between nodes. In fact, an ad hoc network has the capability of making communications possible even between two nodes that are not in the transmission range with each other: packets to be exchanged between these two nodes are forwarded by intermediate nodes, using a routing algorithm. Hence, a MANET may spread over a larger distance, provided that its ends are interconnected by a chain of links between nodes (also called *routers* in this architecture). In the ad hoc network shown in Figure I.5, node A can communicate with node D via nodes B and C , and vice versa.

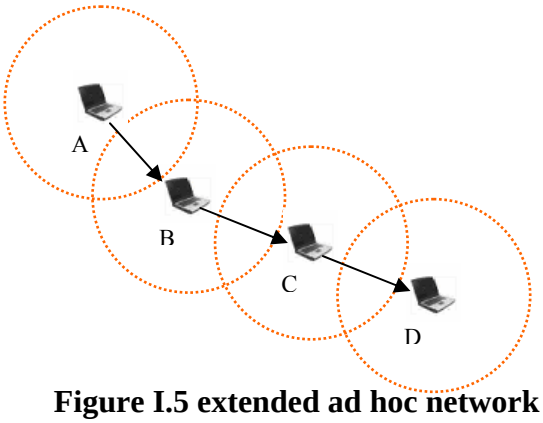


Figure I.5 extended ad hoc network

I-4-1 Benefits of ad hoc networks

WLANs offer four primary benefits:

a- User Mobility

Users can access files, network resources, and the Internet without having to physically connect to the network with wires. Given them the possibility to be mobile and attached to the network in the same time [3].

b- Rapid Installation

The time required for installation is reduced because network connections can be made without moving or adding wires, or pulling them through walls, or making modifications to the infrastructure cable plant. Because the elaboration of an ad hoc network needs a set of nodes having only wireless NIC (Network Interface Controller) [3].

c- Flexibility

Ad hoc networks are also flexible, since it can be deployed in different environment, ranging from battlefields to conferences, trade show, or standards meeting, where temporary network are needed to serve some special needs like files and resources sharing between the participant in the meeting or conference [3].

d- Scalability

Ad hoc network topologies can easily be configured to meet specific application and installation needs and to scale from small peer-to-peer networks to very large enterprise networks deployed on a big area [3].

I-5- Characteristics of Ad hoc network

a- Dynamic Topologies

Ad hoc networking topologies are dynamic in nature because nodes can move unpredictably. Links between nodes can be broken at any time because of arbitrary movement of nodes. This salient feature of ad hoc networks makes it difficult to establish secure key distribution and routing protocols for mobile ad hoc networks [3,4,6].

b- Limited Bandwidth

In ad hoc networks, nodes have to rely on wireless links for communicating with each other. Usually wireless links have less bandwidth than that of traditional wired link due to effects of fading, noise, multiple access and interference conditions. Limited bandwidth can very often be an obstacle for increasing demand of various services in ad hoc networks [3,4,6].

c- Energy Constrained Devices

Most of the nodes participating in ad hoc networks are small portable devices. These devices depend on batteries or other exhaustible means as their power sources. For this reason, it can be very well be possible to launch a denial of service (DoS) on a mobile node by consuming its battery power. Sometime, cryptographic operations that require complex mathematical calculations become difficult with energy constrained devices [3,4,6].

d- Limited Physical Security

In mobile ad hoc networks nodes have very limited physical security then their wires counterpart. As the nodes are exposed and mobile, the possibility of nodes being compromised physically is very high. And the transmission range of the network may exceed the range of the network which exposes it to several attacks [3,4,6].

e- Decentralized administration

In ad hoc network there is no concept of centralized administration attached to wired network. Thus nodes in the network must collaborate in order to accomplish some task like key and authorization right distribution, or rely on an off-line administrative authority to distribute administrative information before the elaboration of the network [3,4,6].

I-6- Applications of ad hoc networks

To motivate the development of ad hoc networking protocols, there are needs to be applications where the properties of ad hoc networking are beneficial. This section will cover some famous applications where ad hoc networks are used but the list isn't exhaustive and other application may exist everywhere when a deployment of an infrastructure network is costly or difficult:

a- Military Tactical Networks

The first application of ad hoc networking was in the military domain. Ad hoc networking enables battlefield units to communicate anywhere and anytime, without the requirement of any fixed infrastructure. The fact that every node forwards packets also provides a robust network. The loss of any one unit will not disrupt the network since there will be other units

that can still provide packet forwarding services. The first known use of these networks is in the gulf war in Irak where USA army had used it to enable communication between their soldiers, to give them more mobility and effectiveness [4,5].

b- Personal Area Networks

The concept of personal area networks consists of interconnecting different devices used by a single person, e.g. a PDA, cellular phone, laptop etc. This kind of network is used to exchange information between devices in home for example, it is accompanied with a services discovering protocol to enable interoperability between devices in the same area [1].

Another example could be when a person holding a PDA comes within transmission range of a printer. If both the PDA and the printer were ad hoc enabled the PDA could automatically get access to the printing services.

c- Sensor Networks

Sensor networks are ad hoc networks consisting of communication enabled sensor nodes. Each such node contains one or more sensors, e.g. movement, chemical or heat sensors [7]. When a sensor is activated it relays the obtained information trough the ad hoc network to some central processing node where further analysis and actions can be performed.

Such sensor networks may consist of hundreds or thousands of sensors and can be used in both military and non-military applications, e.g. surveillance, environmental monitoring etc.

Sensor networks differ significantly from the other types of ad hoc networks described in this section. The most significant difference is the small size, extremely limited power resources and processing power of the sensor nodes [8]. Which means that any protocol developed for ad hoc network must be adapted to sensor network tacking in account these characteristics. An application is described in [9], which consist to use sensors networks to interconnect medicine sensors attached to patients allowing in this way telemedicine.

d- Collaborative Networking

This application of ad hoc networking may be the most intuitive. The simplest example is when a group of people are attending a meeting and need to share information between their laptops or PDAs. If these devices were ad hoc enabled they could dynamically set up a network consisting of the meeting participants and thus enable the sharing of the information and services. These kind of network has a small lifetime so a wired network isn't desired which justify the deployment of wireless networks [4,5].

e- Disaster Area Networks

Ad hoc networking allows the quick deployment of a communication network in areas where no fixed infrastructure is available or where the fixed infrastructure has been destroyed by natural disasters or other events. Thus such networks could be used to improve the communication among rescue workers and other personnel and thereby support the relief efforts [4,5].

f- Public Area Networks

This kind of network is deployed in public area like air port markets or in hotspots. Wireless network allows people to be connected to the existed local area network and profit from the services of the existed network, which allows the presentation of other services like publicity or the e-buy in this area [4,5]. A known application is the location oriented publicity where the system sends publicity and services to any person in the network according to his location in the public area.

I-7- Problematic

In the previous section we have said that an ad hoc network has a decentralized administrative authority, making some tasks relying on a centralized administrative authority in wired network like routing and security unfeasible in the same way in ad hoc network. Thus other mechanisms must be developed to accomplish these tasks by including all or a subset of nodes to collaborate in order to accomplish these tasks. In the following subsections we are going to devote the most known problematic of ad hoc networks:

I-7-1 Routing protocols for ad hoc networks

In ad hoc networks, to ensure the delivery of a packet from sender to destination, each node relay on its neighbouring nodes to forward packets to nodes which are not in its transmission range. Therefore any node in the network plays two roles the first one as an ordinary node and the second one as a router. So each node must run a routing protocol and maintain its routing tables in memory. The problematic in ad hoc network is the dynamic topology of the network, giving for network nodes more liberty to move around the network, resulting on the fast changing in routs, which means that classical routing protocols can't be directly applied for ad hoc network.

Routing protocols for ad hoc networks can be classified into the following categories: reactive, proactive, and hybrid protocols [6,10]. There exist nowadays almost one hundred routing protocols, many standardized by the IETF (Internet Engineering Task Force) and others still at the stage of Internet-Draft. This section gives, for each category, an overview of the most important ones.

a- Reactive protocols

Under a *reactive* (also called *on-demand*) protocol, topology data is given only when needed. Whenever a node wants to know the route to a destination node, it floods the network with a route request message [6,10]. This gives reduced average control traffic, with bursts of messages when packets need being routed and an additional delay due to the fact that the route is not immediately available. The much known reactive protocol is DSR (Dynamic Source Routing)[11], however they exist other like AODV (Ad hoc On-demand Distance Vector routing)[12] and DSDV (Destination-Sequenced Distance-Vector routing)

b- Proactive protocols

In opposition, *proactive* (also called *periodic* or *table driven*) protocols are characterized by periodic exchange of topology control messages. Nodes periodically update their routing tables [6,10]. Therefore, control traffic is more dense but constant, and routes are instantly available. Some examples of these kind of routing protocols are: OLSR (Optimized Link State Routing)[13], OSPF (Open Shortest Path First) and FSR (Fisheye State Routing).

c- Hybrid protocols

Hybrid protocols have both the reactive and proactive nature [6,10]. Usually, the network is divided into regions called clusters, and a node employs a proactive protocol for routing inside its near neighbourhood's region and a reactive protocol for routing outside this region. Two known example of hybrid routing protocols are ZRP (Zone Routing Protocol)[14] and CBRP (Cluster Based Routing Protocol) [15]

I-7-2 Security problematic in ad hoc networks

Ad hoc networks are by nature very open to anyone. Their biggest advantage is also one of their biggest disadvantages: basically anyone with the proper hardware and knowledge of the network topology and protocols can connect to the network. This allows potential attackers to infiltrate the network and carry out attacks on its participants with the purpose of stealing or altering information.

Another problem is the no existence of centralized authority, responsible for distribution of cryptographic keys, or to manage security mechanism like in wired networks. Therefore new mechanisms must be developed to ensure security in ad hoc network. These mechanisms must be based on the collaboration of all or a subset of the network nodes to achieve security services. These mechanisms must take into account the effect that nodes are exposed to different risks like device theft.

I-8- Security in ad hoc network

I-8-1 need of security

The principle of ad hoc networks sounds like a great idea. A dynamic connection between devices that can be used from anywhere and offers limitless business, recreational and educational opportunities appears to be a promising technological advancement towards making our life easier. However, as with conventional networks, security and safety considerations have to be taken into account.

Security in ad hoc network is a persistent need [16,17,18], since the ad hoc networks have not boundaries and the transmission range of the network may exceed the area where the network is deployed exposing the network on new specific attacks, which can't easily detected like eavesdropping.

Also, depending on the application, certain nodes or network components may be exposed to physical attacks which can disrupt the functionality. In contrary to conventional networks, ad hoc network hosts are more often part of an environment that is not maintained professionally. Wireless nodes might be scattered over a large (potentially unsecured) area, where it may pose difficulties to supervise all of them.

Another specialty of ad hoc networks is their heavy reliance on inter-node communication, due to the dynamic nature of the link between the single nodes, to ensure routing, gives large area of attacks on the routing protocols [10].

I-8-2 The security goals

The security goals of mobile ad hoc networks (MANET) are not different from wired networks:

a- Confidentiality

Confidentiality ensures that only authorized persons can have access to certain classified information. Applications that use ad hoc networks like in military operations, have certain information which can be very sensitive. So, disclosure of such information can be very costly and turn into a devastating situation [3,4,5,19].

b- Availability

Availability ensures that the requested service should be available when requested. So, availability opposes Denial of Service (DoS). With denial of service attack an adversary can

also break down important services like key management. So, availability is an important security goal that should be achieved in any kind of ad hoc networks application [3,4,5,19].

c- Integrity

Integrity implies that messages should be un-altered during its transmission from source to destination. Messages can be modified un-intentionally during transmission because of radio propagation impairment. However a malicious attacker can also modify a message intentionally during its transmission [3,4,5,19].

d Authentication

Authentication is the process of identification, that a receiving entity is assured that the message he receives come from a legitimate source. In an ad hoc network, mobile nodes are susceptible to compromise. Without proper authentication, a malicious attacker can impersonate to be an authenticated user and thus can have the full control of the entire network [3,4,5,19].

e- Non-repudiation

Non-repudiation implies that once a message has been sent, the sender can not deny afterwards that it was not he, who sent the message earlier. It is an important security service by which compromised nodes can be detected and isolated [3,4,5,19].

I-8-3 Identification of attacks

Several attacks have been identified against ad hoc networks, some of them are also known in wired network, and other are specific to ad hoc networks. In this section we are going to present a non exhaustive list of attacks.

Network security attacks are typically divided into *passive, active* or *physical* attacks. These classes are then subdivided into other types of attacks [20].

I-8-3-1 Passive Attack

It is an attack in which an unauthorized party gains access to the transmission range of a WLAN and listens passively without modifying the content of traffic. Passive attacks can be either eavesdropping, traffic analysis or impersonation attacks. Typically passive attacks may be the first step to performing active attacks, allowing the attacker to gain some interesting information about the system security.

a- Eavesdropping

The attacker monitors transmissions for message content [20, 21, 22]. An example of this attack is a person listening into the transmissions on a WLAN between two workstations or tuning into transmissions between a wireless handset and a base station.

b- Traffic analysis

The attacker, in a more subtle way, gains intelligence by monitoring the transmissions for patterns of communication [20,21,22]. A considerable amount of information is contained in the flow of messages between communicating parties, which can be used to conclude other important information like IP address used after to perform spoofing attacks.

c- Impersonation

Impersonation attacks are also called *spoofing* attacks [21,22]. The attacker assumes the identity of another node in the network, thus receiving messages directed to the node it fakes.

Usually this would be one of the first steps to intrude a network with the aim of carrying out further attacks to disrupt operation. Depending on the access level of the impersonated node, the intruder may even be able to reconfigure the network so that other attackers can (more) easily join or he could remove security measures to allow subsequent attempts of invasion. Depending on the layer where the identity faking takes place, it can be difficult to prevent it.

I-8-3-2 Active Attacks

An attacker whereby an unauthorized party makes modifications to a message, data stream, or file. It is possible to detect this type of attack but it may not be preventable. Some of these attacks are defined below.

a- Sinkhole attacks

By carrying out a *sinkhole attack*, a compromised node tries to attract the data to it from all neighbouring nodes [21,22]. Since this would give access to all data to this node, the sinkhole attack is the basis for many other attacks likes eavesdropping or data alteration. Sinkhole attacks make use of the loopholes in routing algorithms of ad hoc networks and present themselves to adjacent nodes as the most attractive partner in a multihop route. Effective against sinkhole attacks is the use of multipath routing protocols [23]. Multipath routing protocols send data redundantly, over more then one path ensuring in this way the true deliverance of data; however it doesn't eliminate the eavesdropping problem.

b- The Sybil attack

Malicious nodes in a network may not only impersonate one node, they could assume the identity of several nodes, by doing so undermining the redundancy of many routing protocols [21,22]. Since ad hoc networks depend on the communication between nodes, many systems apply redundant algorithms to ensure that the data gets from point A to point B. A consequence of this is that attackers have a harder time to destroy the integrity of information. If the same packet is sent over several distinct paths (in multipath routing protocols [23]), a change in the packets incoming from one of these paths can be detected easily, thus isolating a possible intruder in the network becomes possible

However, if a single malicious node is able to represent several other nodes, the effectiveness of these measures is significantly degraded. The attacker may get access to all pieces of the fragmented information or may alter all packets in the same transmission so that the destination node can't detect tampering anymore.

c- Denial of Service

A denial-of-service (DoS) attack is one wherein the attacker attempts to render the target network unable to serve its legitimate users. In the wired domain, DoS attacks such as SYN flooding and the Ping of Death, which seek to overwhelm the target network with traffic and force the network servers to crash, have become customary. This type of attack is also effective against wireless networks which are vulnerable to DoS attacks that are not feasible against their wired brethren. Because their signals must travel through the public airwaves rather than in protected cables, wireless networks are extremely susceptible to radio interference, either deliberate or accidental. Accidental interference occurs all too often, owing to the shared, unlicensed nature of the bands in which these networks operate. It is common for a wireless network, or a portion of it, to become unusable when a cordless telephone is operating in the same radio band and in physical proximity to the wireless node, therefore the same principle may be used to attack an ad hoc network [20, 21,22].

d- Data modification

This threat represents a deliberate and unauthorized insertion, alteration or deletion of data in a given system. Example of a specific attack is the *man in the middle attack* where the attacker has to position himself so that all traffic coming and going to the victim goes through him [24]. Then the attacker may perform any of the specified actions; insertion, alteration or deletion. To guard against this type of attack, the use of end-to-end encryption and strong authentication mechanisms with digital signatures are recommended [21,22].

I-8-3-3 Physical attacks

This kind of attacks focuses the equipment used in the network, in order to gain access to the network by using information stored in this device [21,22].

a- Device cloning

Device cloning is the act of presenting false or duplicated credentials to gain access to a system resource; this threat is also called *masquerading* [21,22]. Using strong authentication mechanisms with digital certificates and signatures will protect against most utilizations of this threat.

b- Device theft

Device theft is the physical theft of any given device by an attacker [21,22]. Designing devices or systems resistant to theft is very difficult in general. However all devices should be stored in secure locations when not in use. The probability of device theft is greater in mobile systems (e.g. wireless devices), as physical security is here more difficult to enforce.

I-9- Security mechanism

Several techniques have been presented in the literature trying to solve the security problems in ad hoc network. But the large area of use is reserved to those based on encryption. Encryption plays a significant part in both the WLAN and LAN environments. For wireless users, encryption is particularly important because the wireless platform is often the easiest for an attacker to gain access to LAN, if the flow of data isn't encrypted. Encryption makes the job of an attacker much more difficult and helps protect the users from such exploits. Two kind of encryption can be used in ad hoc network, *Symmetric encryption* (conventional encryption) *and Asymmetric encryption* (public key encryption).

I-9-1 Security Standards for infrastructure based WLAN

a-Wired Equivalent Privacy (WEP)

The current standard for protecting wireless communication from eavesdropping is the WEP algorithm. A secondary function of the WEP algorithm is to prevent unauthorized access to a wireless network.

WEP is a symmetric cryptosystem, which relies on a secret key being shared between the nodes in the network. The secret key is used to encrypt packets before they are transmitted, and an integrity check is used to ensure that packets are not modified in transit. The WEP standard does not specify any techniques for key distribution, but in practice, most installations use a single key that is manually shared between all nodes.

b- IEEE-802.11x

The IEEE 802.11x is a port based standard [2], in which a new user connects to RADUIS server via an insecure port to be authenticated, if the authentication success then the user is authorized to use the secure port otherwise it is excluded. The process of login is as follow:

First the user must introduce its login and password these two identifiers are transferred over the insecure port to the RADUIS server situated in the wired network. If the user is authenticated the a grant response is forwarded to him followed by the symmetric encryption key allowing him to use secured port , otherwise an access denied alert is sent to this user. Any other mechanism can be imagined to enforce security like proactive key update or Public Key Infrastructure to be used over this protocol to enforce security.

c- IEEE 802.11i

The IEEE 802.11i standard is recently developed, and it's intended to improve the WLAN security. Among other things defined in the 802.11i specification are some new encryption key protocols including the Temporal Key Integrity Protocol (TKIP) and the Advanced Encryption Standard (AES). The 802.11i standard will correct the security problems associated with WEP, e.g. TKIP uses an extended 48-bit IV, compared to the 24-bit IV in WEP. It would take approximately 100 years for a key to be reused when using a 48-bit IV under heavy traffic conditions [2].

I-9-2 Security Standards for ad hoc networks

The quest for security in ad hoc network isn't achieved yet, and it is still a great area of debates. Several security management schemes are developed. Some of them based on public key encryption and other on symmetric key encryption and every one of them has its advantages and its disadvantages, in chapter II we are going to present a list of known security management scheme for ad hoc network. And in chapter 3 and 4 we'll present our contribution in the area.

II- Cryptography background

II-1 Cryptography

Is the art of scrambling messages so that it cannot easily be understood by anyone other than their sender and their intended recipient. Cryptography must ensure all security goals (integrity, privacy, authentication and non repudiation), using mathematical equation. Before the apparition of informatics cryptography was only used by government to protect their secrets during wars. However in the information age the cryptography is the best tool to preserve private life, on the Internet. Nowadays cryptography is used to ensure all secure communication on the Internet ranging from the bank and E-commerce transactions to mail encryption.

During the three dedicate ago different kind and cryptographic algorithms have been developed. This can be organized into two categories: *Symmetric* and *Asymmetric* encryption. However we can also add hash functions which are used to obtain message digest used in electronic signature.

II-2 Symmetric encryption

Symmetric cryptography is a cryptographic method employing a single key for both encryption and decryption [25,26]. The use of a single key makes the decryption process a simple reversal of the encryption process. therefore both sender and recipient choose a key of a given lenght and use it to ecrypt and decrypt traffic over the network

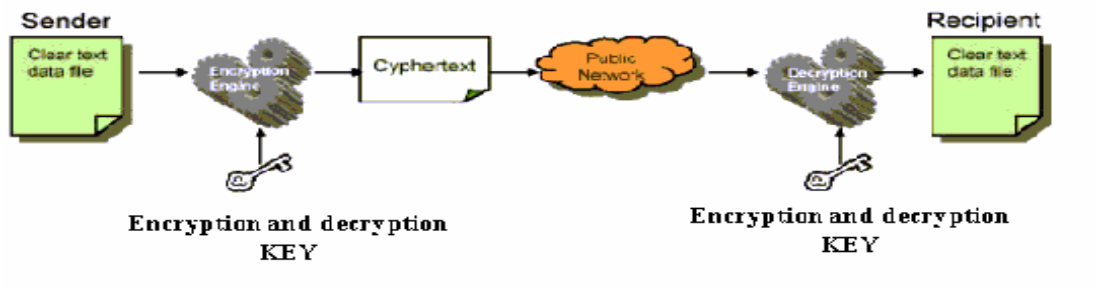


Figure I.6 symmetric key encryption and decyption process

The advantages of such cryptosystem is that, the system perform encryption and decryption fastly compared to other techniques. And it also preserve the same message lenght after encryption, which make it the prefered technique to encrypt traffics over the networks. However it has some inconvinients :

1- it dosn't ensure authentication and integrity, since any malicious entity having the shared key can use it to inject false information, or modify exchanged data over the network.

2- The key is shared among all entities in the network, so an attacker focuses its attacks on one entity in the network to steal the key and gain access to the network.

II-2-1 Example of symmetric algorithms

a- Data Encryption Standard DES

DES has been a worldwide standard for data encryption for more than two decades now. On May 15, 1973, National Institute for Security Technologies (NIST) issued a public request for a data encryption algorithm. This request eventually resulted in the DES implementation. DES was officially endorsed by the U.S. government in 1977 as an encryption standard. Although it was originally developed by IBM (who holds the patent for DES), it has been extensively studied since its original publication. DES is, without doubt, the best-known and most widely used cryptosystem in the world [25].

b- The Advanced Encryption Standard AES

The Advanced Encryption Standard (AES) supersedes Data Encryption Standard (DES) as the new information protection standard defined by the United States to protect certain levels of federal information and communications. The selection process for an AES algorithm began in 1997, and the new standard was finalized in November 2001. The AES standard selected by the NIST specifies the Rijndael algorithm, which is a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. The Rijndael algorithm was submitted by Joan Daemen (Proton World International) and Vincent Rijmen (Katholieke Universiteit Leuven) [25].

II-3 Hash algorithm theory

A hash function has many names, among others; message digest, fingerprint and compression function. A hash function *H* is a transformation that takes a variable-sized input *m* and returns a fixed-sized string, which is called the hash-value *h* (that is, *h = H(m)*). In general *H(m)* will be much smaller in length than *m*; e.g., *H(m)* might be 64 or 128 bits, whereas *m* might be a megabyte or more [26, 27].

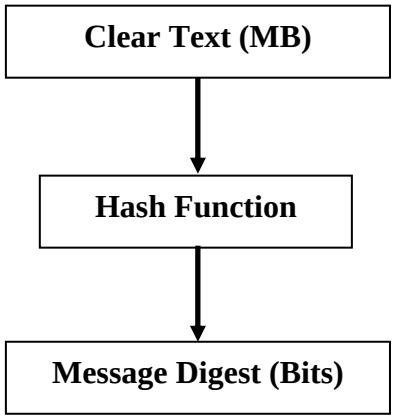


Figure I.7 Hash function

It is theoretically possible that two distinct messages could be compressed into the same message digest, resulting in a collision. The security of hash functions thus requires collision avoidance. Collisions cannot be avoided entirely, since in general the number of possible messages will exceed the number of possible outputs of the hash function. However, the probability of collisions must be minimized. Since hash functions are generally faster than encryption algorithms, it is typical to compute the digital signature or integrity check of some document by applying cryptographic processing to the document's hash-value, which is quite small compared to the document itself. Additionally, a digest can be made public without revealing the contents of the document from which it is derived [3,25,27].

II-3-1 Examples of hash algorithms

a- MD5

The Message Digest algorithm version 5 (MD5), was developed by Ronald L. Rivest at Massachusetts Institute of Technology (MIT). MD5 has been one of the most widely used secure hash algorithms, but due to the threat of both brute-force and cryptanalysis some concerns have arisen [26, 27].

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message called digest.

b- SHA-1

The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a Federal Information Processing Standard in 1993. A revised version was issued in 1995 and is generally referred to as SHA-1. The output after the processing is a 160-bit message digest [26, 27].

II-4 Asymmetric encryption

Asymmetric cryptography is cryptographic methods in which the encryption key is different from the decryption key. With this type of cryptography, the decryption process is not simply an inverse of the encryption process but a totally different mathematical transformation. Asymmetric cryptography is a very powerful technology because a key need not to be shared between two communicating parties.

As shown in Figure I.8 the sender can encrypt data using the public key of the receiver, which is supposed to be publicly known within a given comunity. In the receiver part just the receiver having the true private key, which is supposed to be private and secret can decrypt the message. In this way we can be sure that there is no person in the middle who can read or modify the message.

Another use of public key encryption is to ensure authentication, this is done when the sender encrypts the message with its private key, so every one in the network can read the message and every one is sure that the message is coming from the pretended sender and no one has altred it. However if this isn't the case the decryption mechanims may fail.

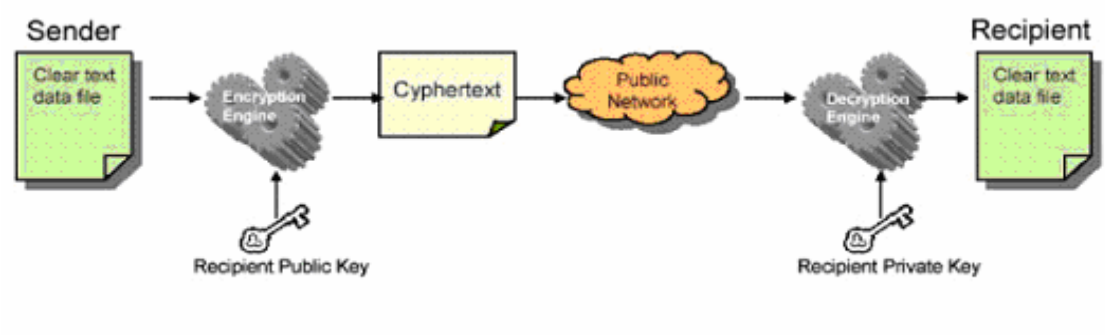


Figure I.8 public key encryption and decryption process

The public key encryption ensures integrity, confidentiality, non repudiation and authentication. However compared to symmetric encryption the public key encryption and decryption is relatively slow and double the size of messages after encryption. Therefore it isn't useful when encrypting big documents. Because of this asymmetric encryption is used for electronic signature.

II-4-1 Example of asymmetric algorithms

a- RSA

RSA is perhaps the most well-known asymmetric or public key technique.

Before we can use the RSA primitives, we must generate public and private RSA keys. The process to generate these keys is as follow [26, 27]:

1- Select two large random prime numbers p and q.

2- Set the values $n = pq$ and $\varphi = (p - 1)(q - 1)$.

3- Pick a random integer e greater than 1 and less than φ so that the only common divisor between e and φ is 1.

4-Derive the integer d between 1 and φ so that $ed \equiv 1 \bmod \varphi$ (that is, the remainder of ed divided by φ is 1).

5-The public key consists of the integers *n* and *e*, and the private key consists of the integers *n* and *d*.

To encrypt an integer *m* with the public key (*n*, *e*) resulting in the integer result *c* you compute

$$c = m^e \bmod n.$$

To decrypt an integer *c* and retrieving the integer *m* with the corresponding private key (*n*, *d*), you compute

$$m = c^d \bmod n.$$

The RSA signature and verification primitives are actually the encryption and decryption primitives in the reverse order. Therefore, to sign or produce a signature *s* of an integer *m*. We encrypt *m* with the private key (*n*, *d*):

$$s = m^d \bmod n.$$

To verify the signature *s*, we decrypt *s* with the public key (*n*, *e*):

$$v = s^e \bmod n.$$

b- Diffie-Hellman

Diffie-Hellman (DH) algorithm was one of the initial public key algorithms used to establish a secret between two parties by exchanging some messages through an insecure channel [26]. In the DH algorithm each of a given two entities A and B chooses a large prime p and a generator g of Z*. Both entity A and B choose a random secret a and b respectively. Then every entity computes respectively g^a mod p and g^b mod p and sends it over the insured channel. When this exchanging is achieved every party compute the secret k= g^{ab} mode p, which may be used to elaborate a secure channel between the two entities.

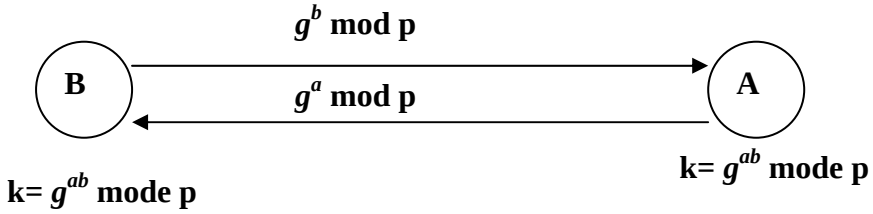


Figure I.9 Diffie-Hellman process

Examining the problem from a different perspective, given only g^a and g^b (that is, we are not given *a* or *b*), it is very hard to compute g^{ab}. Of course, given g^a and *b* (or given g^b and *a*), it is easy to compute g^{ab}. The difficulty of computing g^{ab} given only g^a and g^b serves as the basis for many asymmetric cryptographic protocols.

II-4-2 Threshold Cryptography

Traditional cryptography usually deals with the presence of one sender and one receiver. But, the situation is different in real world scenarios where communications between individual and group and communications between groups are occurring frequently. In some scenarios, many receivers and senders need to share the power of a cryptosystem. The main

motivation of Threshold Cryptography is to develop techniques where multiple sender or receiver wants to share the power of a cryptosystem.
In a (k,n) threshold cryptography scheme:

- n parties share the ability of performing a cryptographic operation.
- Any k of those n parties can perform the operation jointly
- Any k-1 or less parties cannot perform the operation.

a- Secret Sharing

Secret sharing allows a group of users to share a secret, such that all the shareholders can get together and recover the secret and it is unfeasible for less than the total number of users to recover or reconstruct the secret [26]. Thus, in a (k,n) secret sharing scheme a secret s is shared among n parties satisfying the following requirements:
1- Availability: greater than or equal to k parties can recover s.
2- Confidentiality: less than k parties have no information about s.

b- Shamir's Secret Sharing

Shamir has proposed a (k,n) threshold secret sharing scheme based on polynomial interpolations [26].
To share a secret S among k parties, we do the following steps [26]:
i. Let S be the secret chosen from Zp, p prime.
ii. Select a random polynomial $f(x) = f_0+f_1(x_1)+ f_1(x_1) +\cdots f_{k-1}(x_{k-1})$, under the condition that f_0= s and $f_1, f_2,\ldots f_{k-1}$ are chosen randomly from Zp.
iii. For all, $i \in [1,n]$, distribute the share s_i = (i, f(i)) to the *i*th party.

Once the secret has been shared, it can now be reconstructed from every subset of k shares by the Lagrange formula:

$$f(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j\neq i}^{k} (x - x_j)/(x_i - x_j) \bmod p$$

In summary, Shamir's secret sharing scheme is perfectly secure, flexible, and efficient.

c- Proactive Secret Sharing

In a proactive secret sharing scheme, the servers generate a new set of shares for the same secret from the old shares without reconstructing the secret. Proactive secret sharing approach can enhance the security of a service using shamir's secret sharing, by updating the sub shares of each entity after a given period. This technique is innovated from the assumption that an attacker has the possibility to compromise k-1 server after a given period but he can't compromise k server before this period. Therefore updating the sub shares before the expiration of this period keeps the system security out of risks of discovering the secret. The update of the sub shares is done as follow [26]:

- Collaborative generation of the update polynomial *f(x)* (previous section).
- Distribution of the update polynomial to all network nodes.
- Update the shares by every node in the network by adding this polynomial to the old sub share.

II-4-3 Problematic

As we have said public key encryption ensures both integrity, confidentiality, non repudiation and authentication. However the use of public key encryption for authentication must be accompanied with the use of another mechanism ensuring the dependency of the public key of any person and its identity as known in the community. This mechanism is called certificate and it's much known in wired networks, however it is still questionable if it can be used in ad hoc network or not.

Another problem is the use of public key encryption in ad hoc network since the nodes in the network have less computing resources, so such cryptosystem must be avoided when encrypting traffic over the network. And it's suitable to use symmetric key encryption to accomplish this.

II-4-4 Security and cryptography

Before acheiving the section of cryptography we must mentionne that ccryptography is not security. In particular, application security is more than just cryptography. Strong cryptography is usually a prerequisite for secure applications, but the mere use of cryptography cannot guarantee that an application will be secure. So any cryptosystem can't ensure security without a strong underlying key management.

II-5 The Digital Signature

A digital signature is the encryption of a message with a private key. However it's unlikely to use this mechanism when the message exceeds the megabyte, because asymmetric encryption consumes lot of resource and it isn't desired to accomplish big messages encryption. To ensure signature we must proceed as follow:

Using a hash function we compress the document, obtaining in this way the digest of this message which is fixed in length for any size of input text. Then we encrypt this digest with the private key of the sender Site A. Then the sender sends both the text and its digest over the network Figure I.10.

A reversed process of the digital signature is used to verify the signature when received in site B Figure I.10. The receiver computes the digest of the text, and then he decrypts the received digest using the public key of the sender. The verification is considered as successful if the two digest are the same. However if the digest are different this means that the message has been altered.

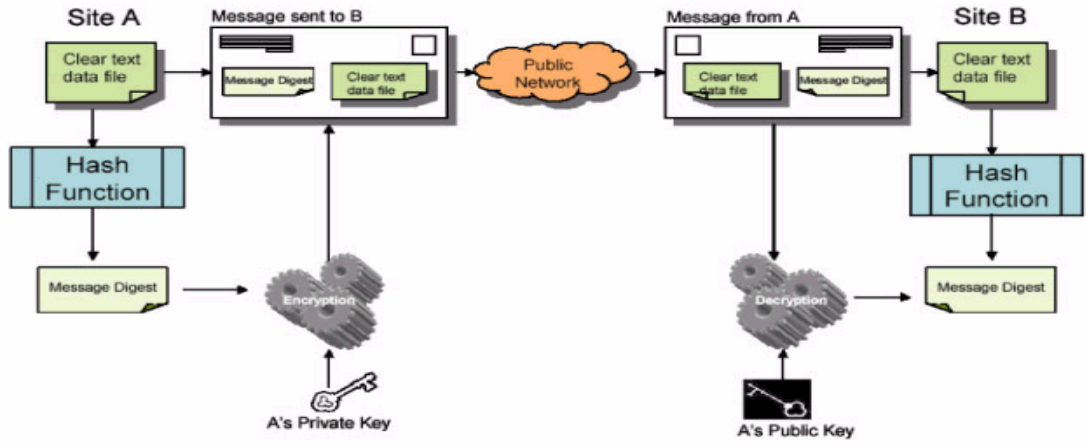


Figure I.10 The digital signature process.

The digital signature is widely used to sign certificate by a certificate authority, giving for users the possibility to verify the validity of this certificate using the public key of the certificate authority, which is supposed to be publicly known in a given community.

II-6 Public Key Infrastructure

In order to protect critical applications and data and comply with new regulatory requirements, public key infrastructure (PKI) has recently increased in popularity for use in the banking, financial, and health care industries and in areas where the protection of proprietary data is imperative [2].

A PKI is a set of technologies that enables an organization to ensure that similar levels and forms of trust that exist in the physical world are implemented in the digital world. It includes the hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke certificates [2].

II-6-1 The Digital Certificate

A digital certificate is an electronic document that binds pieces of information together to include name, serial number, expiration dates, copy of the certificate holder's public key, and the digital signature of a trusted third party called Certificate authority CA which digitally signs and therefore certifies both pieces of information. If you believe that the signing entity is honest and that its private key has not been compromised, then you can safely assume that the signing entity believes that the public key belongs to the named entity [2,27].

In order to validate the signature of the signing entity, you need the signing entity's public key. That public key is often stored in a self-signed certificate, a certificate digitally signed by the same entity whose public key is contained within the certificate. A certificate is also characterized by a validity period after which the certificate is automatically revoked. Otherwise it must be renewed before the expiration of the validity period [2,27].

II-6-2 Public Key Infrastructure components

a- The Certificate Authority CA

The CA issues the actual certificates and implements the defined policies and procedures on how those certificates are to be utilized. A CA generates updates, manages certificates, signs certificates, stores users' private keys, generates and publishes the Certificate Revocation List (CRL), and cross-certifies other CAs. Depending on the application or role, any organization that has the ability to verify the binding between a public key and an entity can be a CA. PKI must also provide the necessary components permitting users to verify, register, renew old certificate and revoke compromised ones. These components must also be implemented in simplest manner in order to facilitate task to users [25, 27].

b- The Registration Authority RA

The Registration Authority (RA) is an optional component in the PKI and is a subordinate server to which a CA can delegate management functions. The RA is used to make dependency between the true identity of a given user and its future pair of keys. Therefore any new coming to PKI managed community need firstly to be registered with the corresponding RA, by giving all information (name, mail, address…etc) needed by this community to register a new user, then the RA performs the needed operation to register this user with the

CA which publish its certificate in its directory after creating the pair of keys and adding the corresponding validity period. Thus a RA plays the role of intermediate between the users and the CA which may reduce the overhead on the CA [25, 27].

c- Certificate Distribution and publishing

The CA is the trusted third party and must have a means to distribute certificates, so users and applications can use them. A directory is a certificate repository that stores certificates so applications can retrieve them on behalf of users [25, 27]. The Lightweight Directory Access Protocol (LDAP) has become the directory of choice for many PKI systems. LDAP is popular because it can support a huge number of users, is very scalable and distributed, responds efficiently to search requests, and is an open standard. Directories are an efficient means for certificate storage and retrieval within a PKI system. The CA populates its directories with certificates and CRLs. The directory can then be used by client applications to retrieve the certificate based on a parameter such as name or e-mail address. Clients can also check the CRL to determine whether an individual certificate is revoked or not [25, 27].

e- Certificate Revocation List CRL

Many certificates have a long lifetime, and it is possible for certificates that are no longer trustworthy. The CA must revoke untrustworthy certificates [25, 27]. Reasons that a certificate may be revoked include a compromised or stolen private key, a forgotten user pass phrase, a user who resigns. Another parameter causing certificate revocation is the expiration of its validity period, since each certificate is valid only for a limited period after which it must be revoked.

The revocation status of a certificate must be checked before each use, and the users and applications must be informed that the continued use of the certificate is no longer considered secure. This requires that PKI must incorporate some type of revocation system to permanently publish information regarding the status of each certificate in the system.

II-6-4 X509 standard

X.509 v3 is the current standard of public key certificates that has been widely used [5,6,19]. As we can see on the Figure I.11 each certificate contains a set of information concerning the owner (name, subject, public key…) and the issuer (CA). It contains also a serial number which can be the same for different CA, validity period after which the certificate is automatically revoked. All these information are then signed by the CA's private key, and delivered to the end user. The signature is the last field, allowing the verification of the information contained in the certificate.

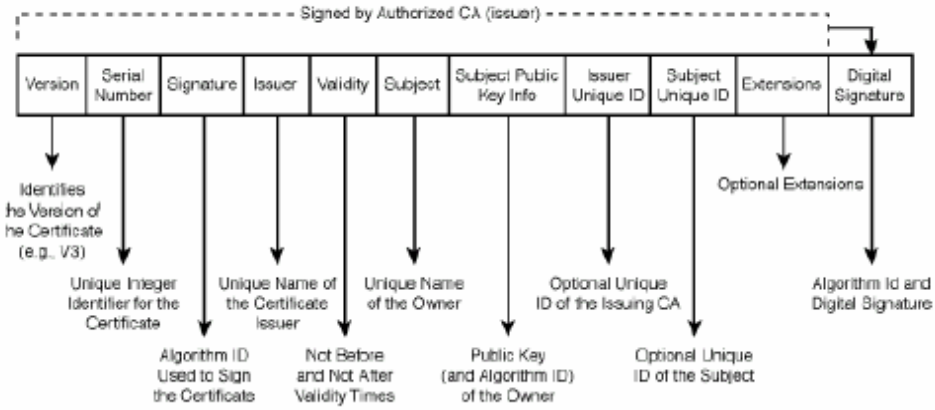


Figure I.11 X.509 v3 Certificate Structure

II-7 PKI in ad hoc networks

As we have said PKI is an infrastructure for managing digital certificates. The most important component of PKI is the CA (Certificate Authority), the trusted entity in the system that vouches for the validity of digital certificates. The success of PKI depends on the security and availability of the CA to the principals in a system (or the nodes in a network) since a principal must be able to correspond with the CA to get a certificate, check the status of another principal's certificate, acquire another principal's certificate, and so on. PKI has been deployed for wired networks [5,6,19] and some infrastructure-based wireless networks [19]. Since good connectivity can be assumed in these networks.

However, it is unclear if such approaches can be extended to ad hoc networks due to the infrastructure-less nature of ad hoc networks inhibits guaranteeing any kind of connectivity. Another serious problem present in ad hoc networks is the increased physical vulnerability of the nodes themselves [28]. Considering that many ad hoc networks will be deployed with mobile nodes, the possibility of the nodes being captured or compromised in a hostile environment is higher than in wired networks with stationary hosts. Mobile nodes in infrastructure-based wireless networks have the same vulnerability, but they can rely on the infrastructure for detection of compromised nodes. Since there is no stable entity in an ad hoc network, ad hoc nodes cannot enjoy such conveniences.

Several proposed solutions for providing PKI for ad hoc networks have been proposed in literature trying to address the increased vulnerability of the mobile nodes by employing techniques to distribute the CA functionality across multiple nodes, generally using threshold cryptography, and other using cluster based architectures [5, 6]. But no one has addressed the full characteristics of ad hoc networks, resulting sometimes on vulnerable implementation or unpractical solutions due to overhead imposed by the implementation. In the next chapter we are going to present some of these techniques.

Chapter II

State of the art

This chapter gives a brief state of the art of security key management schemes for ad hoc networks. Since the number of solutions given in this area is very great we can't present them all, therefore we try to give the most known solutions from which are derived new solutions. This chapter is mainly divided into three sections.

In the first section we give a brief definition of key management for ad hoc networks, and the evaluation criteria on which we judge if a given key management is considered as effective or not for ad hoc networks.

The second section will be consecrated to key management schemes based on the Public Key Infrastructure. Four key management schemes are presented each one is followed by an analysis section giving its possibility to be used in ad hoc networks and its vulnerabilities it they exist.

In the third section we are going to present three different key management schemes, which aren't based on Public key cryptography. Each key management scheme is based on a stand alone technique making it different from the others, giving it the possibility to be used or not for ad hoc network.

1- Key Management

As its name implies, *key management* is concerned with the way that applications handle cryptographic keys. Key management addresses issues such as key generation and distribution, key agreement, and key lifecycle.

Any successeful key management must garanties the following aspects:

-The same cryptographic key should not be used for multiple purposes. For example the same symmetric key should not be used for symmetric encryption and symmetric message authentication code. Multiple usages may inadvertently leak information useful to an attacker.

-Multiple parties should not share the same key. Although this principle may seem intuitive (because the more parties that know a key, the more likely it is for an attacker to learn the key).

-Keys should be changed over time. Regularly changing keys limits an attacker's ability to learn about the key, by placing a time period on a key's usability. This period of time vary according to the encryption algorithm used, and the lenght of the key.

2- Evaluation Criteria

Establishing a secure key management protocol in mobile ad hoc network should satisfy the evaluation criteria [29] discussed in this section. Though, it is not always possible to fulfill each and every requirement, but a key management must ensure the maximum of the following criteria:

2-1 Confidentiality

The shared key that is established using a key management protocol must be inaccessible to any un-authorized party. Using a single shared key is also vulnerable from a security point of view, because un-authorized access to this key can lead the whole system to be insecure. Ad hoc networks are prone to eavesdropping, so identity information and public keys of the nodes should also be encrypted to protect them against traffic analysis.

2-2 Availability

A key management protocol must ensure that the authenticated nodes get the service available when needed. A key management protocol must not rely on a centralized server for key distribution, because in that case it will create a single point of failure in the network and thus limits the availability of the service. The security mechanism should not allow the participating nodes to perform un-necessary operations because this can affect the availability of the service.

2-3 Freshness

It is an important criterion for an ideal key establishment process to generate fresh keys. Participating nodes in a network must change their keys over time using proactive key update, so a key management protocol must ensure that each node in the network has the freshest key.

2-4 Scalability

Ad hoc networks can extend from small to very large networks. As the network size increases, complexity in managing keys also gets complex. For a smaller domain it is better to go for a simpler design rather than a complex one. Any key establishment process has some cost related parameters like number of cryptographic operations, number of messages transferred…etc, which grows rapidly as the networks size increases. A key management protocol should have a better scaling property that can deal with a potential growth in network size.

3- PKI based key management schemes

3-1 Partially Distributed Certificate Authority

This solution proposed by Zhou and Hass [30] uses a *(k, n)* threshold scheme to distribute the services of the certificate authority to a set of specialized server nodes. Each of these nodes is capable of generating a partial certificate using their share of the certificate authority (CA) signing key K_s, but only by combining *k* such partial certificates can a valid certificate be obtained.

3-1-1 System structure

In this solution the author supposes that every node has its public and private key, generated by an off line administrative authority (dealer).
The system contains three types of nodes:

- Client nodes: are the normal users of the network.
- Server nodes: are parts of the CA, every one among them having a partial signing key, which may be used to cooperate with other server nodes to sign certificate and accomplish CA services.
- Combiner nodes: are server nodes having the task to combine *k* partial certificate and generate a valid one, for client nodes.

Before starting, the dealer generates the private and the public key of the CA. Then it distributes the public key to all network nodes. It also divides the private key on *k* shares using Shamir's secret sharing scheme, and distribute it secretly to the *n* servers in the network.

In this scheme the author supposes the availability of the *k* server in the neighbour of every node in the network. So every node needing the CA services (verification, renewal…etc) must contact at least *k* servers in its neighbourhood by sending them a request involving the certificate to be signed. Every server node signs the certificate with its sub share [29,30] and sends it to the combiner who combines these partials to get a valid certificate [29,30].

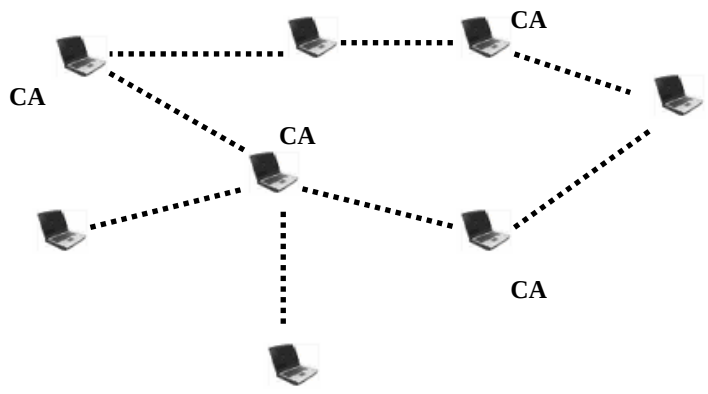


Figure II.1 Partially Distributed

3-1-2 Certificate services

a. Certificate Issuing

Before any node may join the network it must first obtain a valid certificate from an off-line dealer, this certificate is valid since it was signed by the dealer, so a node can use it to join the network. When joining the network packet sent by this node are forwarded because its certificate is valid and verified using the CA's public key which is supposed to be known to all network nodes.

b. Certificate Renewal

The certificates are only valid for a certain amount of time, therefore they need to be renewed before they expire, and they may also be renewed when some of included data has changed (name pair of keys…). When a node wishes to renew its certificate, it must launch a certificate renewal request. The request is sent to a minimum of *k* server nodes. If the request is granted, each of these *k* server nodes generates a partial certificate with a new expiration date. These partial certificates are then sent to a combiner, which could be one of the *k* servers, which then combines the partial certificates and generate a valid certificate [5].

If any of the servers are compromised they may generate an invalid partial certificate. Thus the certificate produced by the combiner will also be invalid. So the operation must be repeated until a valid certificate is obtained.

c. Certificate verification, revocation

The server nodes are responsible for storing the certificates of all nodes in the network. To be used for verification and revocation. So the servers must have at least two directories, one is used to store valid certificate, this directory is consulted when a node need to verify the validity of any other certificate. The second directory is used to store revoked certificate, which aren't valid due to the expiration of its validity time or because its holder has got a malicious comportment.

This service requires that all nodes must register their certificate with the servers when they initially join the network. The servers must also have a mechanism of synchronizing their certificate directories in the case of updates and renewal, or revocation [5].

This service is accomplished in the same way as registration or renewal, by sending a request to neighbour nodes.

d- System Maintenance

The maintenance of the certificate authority consists of updating sub shares, to prevent them from compromising, so at periodic intervals the servers update their sub shares of the *CA*'s private key. By using the proactive secret sharing scheme described in the previous chapter. To accomplish this, random update function is generated by the servers and added to the original function to obtain a new function, and then every server calculates its own share.

During the share update a malicious server can potentially launch a denial of service attack against the distributed CA by generating invalid sub shares, in this situation the author proposes to use the verifiable secret sharing, to verify sub shares validity [5].

Due to the highly dynamic topology of ad hoc networks, all servers might not be connected during the share update. Therefore, new mechanisms must be in place to handle such situations. An example would be if the network is segmented into more than one group, in which each part may update their shares independently of each other.

3-1-3 Analysis

In this section we try to analysis this solution regarding its effectiveness, feasibility and security:

1. This scheme need an administrative authority to generate and distributes the pair of key for all the network nodes, which isn't always the case in ad hoc network where nodes must operate without any administrative authority.
2. This scheme supposes the existing of at least *n* nodes in the network which isn't always the case, because the number of node may decreases and becomes less than *n*.
3. It need an off-line dealer to distribute and signs certificates, which isn't a component of ad hoc network. So it is not present to accomplish other initialization if it is needed for generating new pair of keys if the nodes have not the capability to do this
4. Several denial of services attacks can be executed, when executing proactive share update, or using *CA* services by generating wrong information for every new request.
5. The number of messages exchanged between nodes combiners and servers, add a great overheads to the network.
6. There is no dependency between the certificate and their holder, so impersonate attacks are always possible, by using valid information and certificate by malicious persons.
7. This scheme doesn't treat scalability. Because the predefined value of *k,* and *n,* are defined for a certain size of the network. So if the number of nodes increases the *n* servers may be incapable to accomplish *CA* services and become busy.
8. The author treats availability by giving an appropriate value to *k* and *n.* but the availability isn't guaranteed because servers or nodes can't be always in the same area, and can be disconnected from the network due to their movement.
9. The author doesn't define a recovering mechanism whenever a set of server nodes leave the network, to replace them.
10. The number of operations executed by every server node and complexity of this operation increases the resources consumed by every server node, which causes battery drain.
11. It is not a practical solution to be used for securing routing protocol, because the mechanism of verifying signature is done frequently which may overhead the network due to the set of messages exchanged during the verification. Because every routed packet must be validated by consulting this *k* severs, to verify the signature contained in this packet.

    In other hands, this solution ensures a great threshold of security:
    1. It distributes the CA services over k servers which minimizes the risk of compromising them.
    2. The use of proactive key update excludes any long term attacks to compromise one of the k severs.

3-1-4 Examples of use

In [29] the author supposes that nodes in a network are not always with the same capabilities and the speed of movement, so he proposes to employ *(k, n)* threshold scheme by distributing CA functionality over specialised nodes based on the security and the physical characteristics of nodes. He also proposes to use a selection protocol to select a set of server nodes called MOCA (MObile Certificate Authority). With this mechanism he tries to give a solution to scalability, and he also provides availability by selecting less mobile nodes as servers. He gives an example of a battlefield were MOCAs are deployed on vehicles and other powered nodes. He also tries to minimize the overhead due to messages exchange by using multicast, so messages are sent only to servers which minimises overhead due to flooding.

In [31,32] the authors use clustering concepts to divide the network on a set of small groups called clusters, and they choose one of cluster members to do some specials tasks, which is called cluster-head. In these two proposed solution CA authority is distributed among these cluster-heads. This ensures availability because any node in the network is

always in contact with its cluster-head which guaranties availability. Scalability is also guaranteed due to clustering, by creating new clusters.

3-2 Fully Distributed Certificate Authority

This solution is first described by Luo and al in [33] and analyzed in [34] and [35]. Its uses a *(k, n)* threshold scheme to distribute CA authority between all nodes in the network. It also uses verifiable and proactive secret sharing mechanisms to protect against denial of service attacks and compromise of the certificate signing key.

3-2-1 System structure

One of failure point of the previous solution is that the predefined values of *k* and *n*; may not be always the preferred value for all sizes of the network. So it must be changed whenever the number of nodes in the network increases or decreases. Therefore the author in [33] tries to solve this problem by distributing the signing key among all the network nodes. So he proposes to uses a *(k, n)* threshold scheme, where *n* here is the number of nodes in the network.

An off-line dealer (administrative authority), divides the CA private key on *k* sub shares, using Shamir's secret sharing scheme, and distribute them to every node in the network. So the capability of CA is distributed among all network members. And every CA operation must be executed by a coalition of *k* nodes.

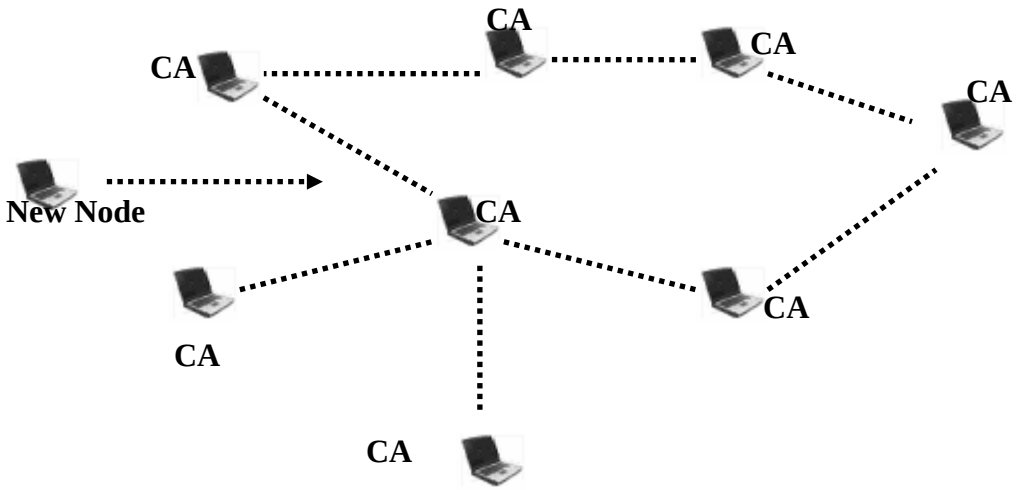


Figure II.2 Fully Distributed Authority

The availability of the service is based on the assumption that every node will have a minimum of *k* one-hop neighbours and that the nodes are provided with a valid certificate prior to their joining the network (dealer).

3-2-2 System Bootstrapping

During the system bootstrapping phase, the administrative authority responsible for the ad hoc network initializes the first *k* nodes. The initialisation includes providing the nodes with their own certificates *cert*, the *CA*'s certificate $cert_{CA}$ and their shares of the CA's private key K_s.

The dealer is the only entity who has access to the certificate signing key K_s and he can therefore issue the initial certificates.

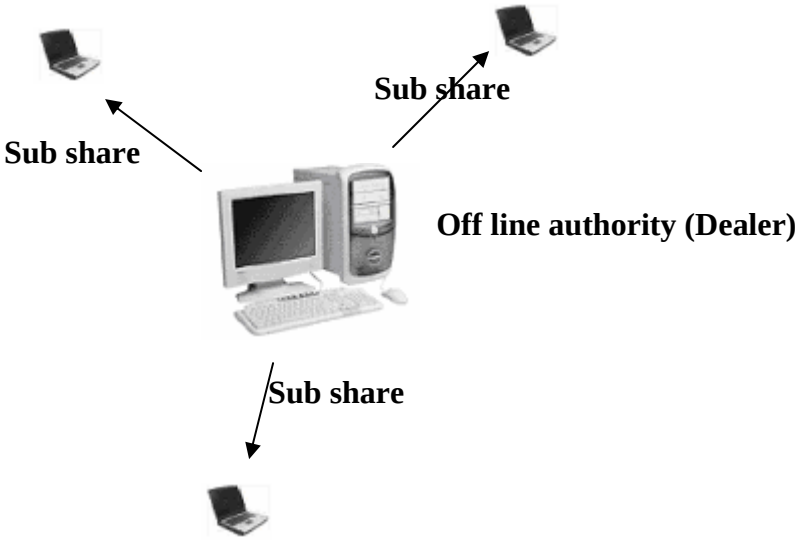


Figure II.3 Initialisation of nodes by the dealer

3-2-3 Share Initialization

As we've said sub shares and certificate are delivered by the dealer, however this dealer isn't part of the network and it may be missing when new nodes join the network. Therefore any new joining node must rely on server nodes to get its sub share and its certificate. So a new joining node must rely on its neighbourhood to be incorporated in the CA, the node need only to collect k such partial shares and combining them together.

The new node broadcast a joining node to a coalition of *k* nodes to be initialised which must compute the corresponding sub share of this node by using Shamir's secret sharing scheme. However the joining node can only be allowed to know the value of the sum of the *k* sub shares, not the value of the sub shares themselves, which are kept secretly by their holders. To allow this the author proposes to shuffles the sub shares before sending them to the node. So every pair of nodes i and j among the coalition of the k servers securely exchanges a shuffling factor $d_{i,j}$. One of the pair adds $d_{i,j}$ to its partial share, however the other subtracts $d_{i,j}$ from its partial share. When summing this sub shares the combiner nodes can get the true sum of the sub shares without knowing the value of any sub share in the coalition [33].

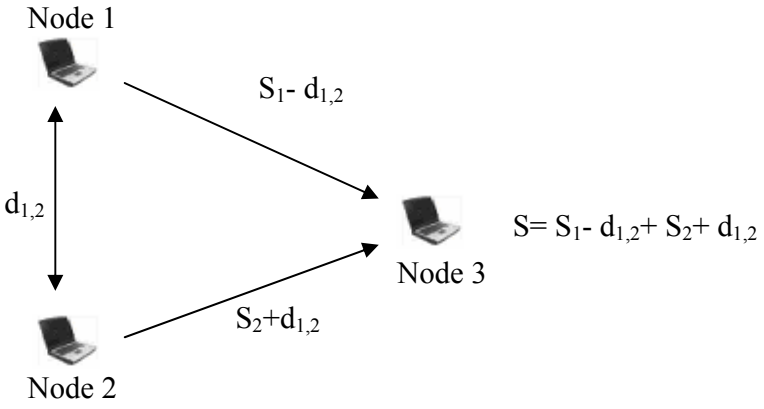


Figure II.4 Initialisation of new nodes

When getting the sum of sub shares the new node can get its sub share using the following equation:

$$S_i = \sum_{j=1}^{k} Sj$$

3-2-4 Share Update

The author supposes that an attacker can compromise a node in a given period, and then he can do the same thing with *k* nodes thereby be able to reconstruct the shared secret (signing key). So he proposes the use of proactive secret sharing, to protect sub shares from long term attacks, which consists on updating the sub shares after a given period.

During the share update phase the following three steps are performed:

- Generation of the update polynomial *f(x),* like it was mentioned in previous chapter.
- Distribution of the update polynomial to all network nodes.
- Update the shares by every node in the network by adding this polynomial to the old sub share.

a- Certificate Issuing

It consists of the generation of a new certificate, since this solution is based on the assumption that all nodes have been initialized, registered, and issued with a valid certificate before they join the network, by a third party (dealer). The distributed CA never issues new certificates; it only manages certificates once they have been initially created by the dealer.

b- Certificate Renewal

When a node *i* wishes to renew its certificate *cert* it requests a certificate renewal from a coalition of *k* neighbours. Each node *j* in the coalition checks that the old certificate is valid, by consulting its own revocation list.

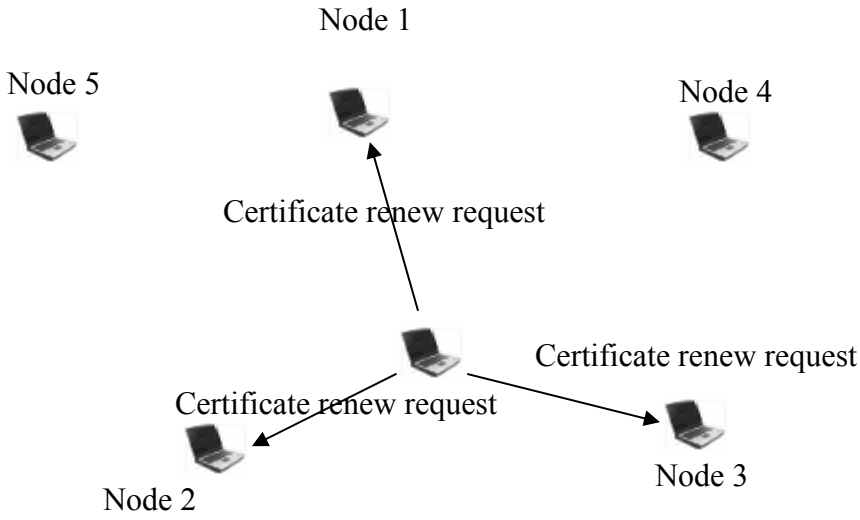


Figure II.5 Certificate renewal process

Every node *j* in the coalition of *k* nodes signs the updated certificate, with its partial share and sends it to node *i*.

$$CERT_j = (cert)^{S_j} \bmod P$$

Upon receiving k partial certificates {$cert_3$, $cert_2$ …$cert_k$}, from the coalition, node i combines them together by multiplication to generate a candidate certificate CERT':

$$\text{CERT}' = \prod_{j=1}^{k} CERT_j$$

The candidate certificate CERT' is different from the real certificate CERT, thus the node i uses the following algorithm to compute the true certificate CERT:

Inputs - CERT' →candidate certificate
- cert →old certificate.
Out puts CERT → renewed certificate.
$Z=(cert)^{-p} \bmod p$
J:=0;
y:=1;
While j<k do
y=y.z mod p;
j:=j+1;
if($cert=y^{Kp}$) then
break;
endif;
endwhile;
Output cert=y;
End.

d- Certificate Revocation

The certificate revocation mechanism is based on the assumption that all nodes monitor the behaviour of their one-hop neighbours and maintain their own certificate revocation lists. If a node discovers that one of its neighbours is misbehaving it adds its certificate to the CRL and floods an accusation against the node, but the accusation is flooded in a limited area, the reason of this is that the certificate is revoked automatically after a given period, due to the expiration of its validity period, so the purpose to use this mechanism is to minimize the effect of flooding messages. Any node receiving such accusation first checks its CRL to verify that the accusation didn't originate from a node whose certificate has been already revoked, or previously accused, because these nodes are not trusted. Otherwise every node receiving this accusation marks the node as suspect. When a threshold of legitimate accusations, i.e. *k* accusations, against the same node is received the accused node's certificate is revoked [33].

3-2-5 Analysis

In this section we'll give our point of view on this solution, since this solution gives a good support of security due to secret sharing among the whole of the network, and the proactive secret update, it presents some problematic:

1. This scheme is not suitable for network where leaving and joining network are very frequent. So the network is overheaded by joining requests to initialize sub shares, which consumes lot of computational resources.
2. If *k, n* are not modifiable then we get a problem when the number of nodes decreases, to be less than *k.*
3. In this scheme every node is forced to be part of the CA, and it may exist in the network nodes with limited capabilities which can not accomplish this task.
4. An additional overhead due to the proactive secret sharing scheme.
5. Due to the time taken by the execution of the proactive secret sharing scheme, we can find a segment of the network which has not updated its sub shares, whenever other segments have already done this.
6. Can not be used to securing routing protocol, where CA services are frequently needed, because in this situation it takes lot of time.

3-3 Self Issued Certificates

This solution is proposed by Hubaux and al in [36] provides a public key management solution similar to PGP (*Pretty Good Privacy*) [37] in the sense that certificates are issued by the users themselves without the involvement of any certification authority.

Unlike the previous public key based solutions presented above, this one is intended to function in spontaneous ad hoc networks where the nodes don't have any prior relationship, and without the existence of any CA, in the sense that certificates are issued and signed by their holders.

This solution is used in the same way as described in PGP, however in the case of ad hoc network humans are replaced by mobile nodes.

3-3-1 System structure

In PGP scheme the certificate aren't issued by some trusted third party (CA). Instead each user has the capability of issuing certificate of other users, based in its knowledge on this person, making this scheme based on human relationship which is robust against impersonate attacks trying to spoof identity of legitimate users.

In the beginning every new coming to the community creates its certificate and signs it with its own private key, then when he wants to communicates with any other person he must exchange with him its certificates. Then every one between them signs the certificate of the

other and saves it in its directory; to say that this certificate is authenticated by him so it is valid.

In order to obtain the certificate of any other person in the community the user must find a valid chain to this person. Thus he contact neighbour users, who he has already validates their certificate to obtain the needed certificate, then every one of them searches the wanted certificate in his directory, if he finds it then he sends it back to the demander, otherwise he do the same thing with his valid neighbours. In this manner a trusted chain is constructed to the desired person (node).

Figure II.6 illustrates a simple example of how PGP works. Omar has exchanged its certificate with Ahmed, so Omar and Ahmed trust each other. And Mohamed has done the same thing with Omar.

In this way Omar trusts Ahmed and Mohamed. Mohamed and Ahmed don't trust each other but they trust Omar. So when Mohamed wants to obtain the certificate of Ahmed, he'll contact Omar to obtain it. Mohamed considers any certificate coming from Omar as valid, so when he gets the certificate of Ahmed from Omar this means that this certificate is valid. The same thing is done by Mohamed to authenticate Ahmed.

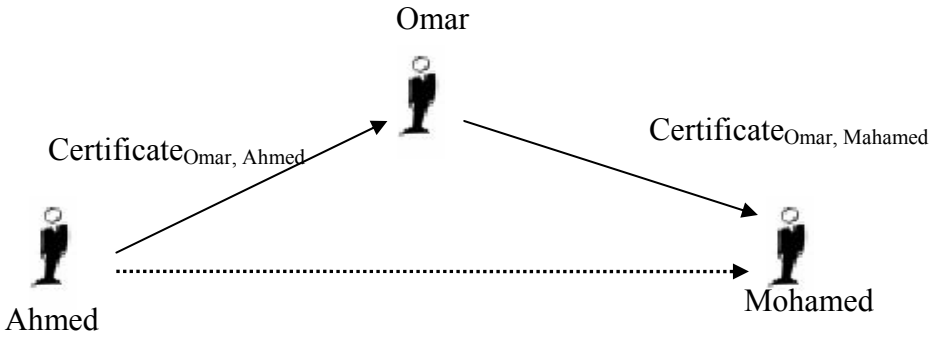


Figure II.6 Certificate chain

In PGP every user keeps a number of directories where he stores the authenticated certificates according to the degree of trust given by the user to their holder When two users wish to authenticate each others, they try to find a certificate chain using only the certificates stored in their directory, if they can't find any chain so they must contact their neighbours in order to find a trusted chain.

An other property of this solution is when a certain person in the network signs a great number of certificates, and it is trusted by the majority of nodes or persons in the network, so any new coming to the network may contact this person to sign its certificate, and obtains other valid certificate from this person, which may minimize the time needed to obtain chains to other persons in the network without contacting any supplementary persons.

If the user don't find any chain to the desired node using his directory he must use the certificate selection algorithm proposed by the authors called Shortcut Hunter algorithm. The Shortcut Hunter algorithm is based on a phenomenon known as the small-world phenomenon and it only provides a probabilistic guarantee of obtaining a certificate chain as described above. The small-world phenomenon supposes to obtain a valid chain to any user in the world

only by contacting a small number of persons. The theories behind the small world phenomenon and Shortcut Hunter algorithm are beyond the scope of this thesis.

The revocation mechanism is also based on accusation, therefore when a certificate of a malicious need to be revoked. A revocation request is revoked to all neighbours, so when any node having this certificate in its directory delete it and does the same thing with its neighbours until the revocation request has traversed the entire network. The speed of revocation depends on the network state and size.

Certificate renewal is done in the same way as revocation, in the way that renewed certificated is broadcasted over the entire network.

3-3-2 Analysis

This solution is very suitable for spontaneous network where there is no centralised authority to distribute certificates. However, due to this it requires an initial phase during which its effectiveness is limited and therefore it is unsuitable for short-term networks.

The speed of the convergence of revocation requests is not suitable for large network, where a node may use the revoked certificate to perform attacks in other segments of the network. Because the revocation request has not yet reach this segment of the network where the certificate of this malicious node is considered as valid.

If this solution is used for ad hoc network then we can find some isolated segment of the network, because there is no trust relation between the bounder users of each segment.

3-4 SEKEN (Secure and Efficient Key Exchange for Sensor Networks)

This solution is proposed by Karman and al in [38]. In this protocol the authors suppose that each node has a unique Device Identifier DId programmed in the devices during construction, this DId is only used during initialisation then it'll be changed by a temporary Id to avoid analysis attacks that can conclude the DId, because it is used to ensure authentication of new coming nodes.

They use the most powered node as base station, which keeps record of all DIds of all possible nodes that may join the network. This base station is the administrative authority of the system.

They also suppose that the base station is created with a public key known in the entire network, they propose that this key is also programmed into device during initialisation.

3-4-1 The protocol

The authors define two basic operations to initiate new joining nodes.

The first one is used to authenticate node with the base station called key setup phase, and the second operation authenticates node with its neighbourhood called key authentication

a- Key Setup Phase

This phase is initiated by the new coming node; first it sends to the base station a join-network message, including it DId_A and a timestamp (TS) encrypted with the public key of the base station.

$$A \rightarrow BS: P_{BS}(DId,TS)$$

The node also calculates the key K_A that it'll share with the base station and sends its using a Message Authentication Code function MAC.

$$K_A = MAC(DId,TS)$$

After receiving the message from node A the base station decrypts it using its private key, and then it verifies if the received DIs is true. Then it computes the K_A in the same manner.

The computed K_A is used to encrypt the following information between the base station and the new node:

A counter C_A: it's initiated to a random value used later to generate a session key between this node and its potential neighbours, and it's incremented in both the base station and the node after each usage.

A temporary ID_A: it is used only for this network; to avoid the compromising of the original device identifier DId, using long term attacks. This temporary identifier is changed if there is any doubt that it was identified by an attacker, however the original device identifier can't be changed because it's programmed in the device by the constructor.

$$BS \rightarrow A : K_A(C_A, ID_A)$$

Whenever this message is received by node A this means that this node is completely authenticated, and it acts like a gateway for other nodes in the network. When another node B_1 join the network it performs the routine operations as described above, however the exchanged data may forwarded over the gateway assumed to be A. the base station then performs the habitual check to authenticate the node B_1 and compute the shared key with the new node B_1 : K_{B1}.

The base station compute the key $K_{A,B1}$= MAC(K_A, C_A) that it'll shares with the gateway A. and sends a message containing all this information to node B1.

$$BS \rightarrow B_1: K_{B1} (K_{A,B1}, C_{B1}, ID_{B1})$$

The same operations are performed with each new coming node as shown in figure II.7

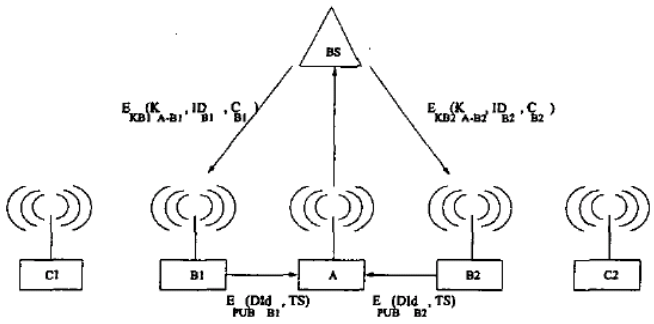


Figure II.7 athentication process with base

b- Key authentication

Whenever the previous steps are achieved each sensor node needs to authenticate its neighbours, for this purpose they use a challenge-response mechanism. Therefore node B_1 generate a random R_A and sends it to A encrypted with the key $K_{A,B1}$ which was got from the base station.

$$B_1 \rightarrow A : K_{A,B1} (R_A)$$

Whenever node A receives this message it compute the key $K_{A,B1}$= MAC(K_A, C_A), and decrypts the message. Then it generates another random R_{B1}, combines it with the R_A and sends it to B_1

$$A \rightarrow B_1 : K_{A,B1} (R_A, R_{B1})$$

Node B_1 decrypt the received message and sends R_{B1} to A always encrypted with $K_{A,B1}$

$$B_1 \rightarrow A : K_{A,B1} (R_{B1})$$

The authentication is successfully if all the preceding operations are achieved without any problem when decrypting or encrypting the random numbers.

Using this mechanism each new coming node authentication all the network sensors.

3-4-2 Analysis

As described in their paper this solution is efficient compared to other solutions, because it has taken into consideration the limitation of power in sensor network. However it isn't suitable for ad hoc networks regarding the following criteria:

-The devices are initiated with DIds which are only used during initialisation, but they don't treat the aspect of changing this DIds when they'll be compromised during the lifetime of the network by theft or any physical attack.

-The authors assume that the public key of the base station is programmed in the base station memory and the ordinary node before deploying them, which make the update of the key difficult when it may be compromised. Since we must update the public key in both devices and base station which is too difficult in large network.

-Each node in the network including the base station shares with each neighbour a session key, which makes collective communication and message diffusing impossible.

-The authentication between sensors is based on symmetric key without proactive key update which is vulnerable to analysis attacks.

This protocol is very useful to authenticate nodes in bootstrapping in ad hoc network to exchange pre-authentication data between a node assumed to be base station or server and other nodes in the network. However it can't be used as a security mechanism in large ad hoc network composed of lot of nodes, because the mechanism of authentication between nodes is very heavy and it's may be launched frequently in ad hoc network, when the topology changes.

4- Other key management schemes

4-1 Secure Pebblenets

This solution proposed by Basagni and al [39] provides a distributed key management system based on symmetric encryption. The solution provides group authentication, message integrity and confidentiality using a set of symmetric keys.

4-1-1 Overview

The solution is intended to be used in large ad hoc networks consisting of nodes with limited processing, storage and power resources. Therefore public key cryptography is not feasible.

To manage security aspects (confidentiality, non repudiation, integrity, authentication) the author proposes to use a set of symmetric keys. He proposes also to manage the network into clusters to simplify key update operation. Thus the network lifetime is divided into three phases:

1. Operational phase: where nodes are exchanging ordinary traffic.

2. Cluster generation phase: in which nodes are negotiating cluster-heads election.
3. Key update phase: During this phase one of the cluster-heads is elected as key manager and is responsible to renew one of the keys (traffic, authentication…) and distributing it to other cluster-heads, which then distribute it to all their cluster-members.

4-1-2 Bootstrapping:

Before joining the network each node maintain three parameters:

- The key update period: specifies the time between two successive key updates. A shortest one ensures higher security and more resources are used, and a longer one signifies a high stability in the network and less security level.
- A unique Identifier ID, to identify every node in the network.
- A statistical average delay Δ, which is used to minimize the risk of multiple nodes becoming key managers.

4-1-3 Cryptographic Parameters

We suppose that all nodes in the network are able to perform symmetric encryption. And maintain the following keys which are used like a seed to generate other keys during the network lifetime:

- Group identity key $K_{GI:}$ used as a seed for other keys generation and for group authentication.
- Traffic encryption key K_{TEK}: used for encrypting ordinary data exchanged between network nodes. K_{TEK} is generated randomly by the key manager.
- Cluster key K_c: it is generated by the cluster head to be used for intra cluster communication; it is different for each cluster.
- Backbone key K_B: used for encryption of cluster head communication. It is used for key manager election.
- Hello key K_H: used during cluster head election.

4-1-4 Cryptographic Functions

In order to ensure several security aspects the author proposes to use the following cryptographic functions:

- One-way hash function: is used to ensure integrity.
- Symmetric encryption algorithm: used to encrypt all exchanged data between nodes.
- Secure key generation algorithm: is used to generate new keys, when performing key update operation mentioned above.

4-1-5 Cluster Generation Phase

It is based on the weight of nodes, which represents its current state of the node regarding the remainder of battery power, distance to other nodes etc. to generate clusters the following steps are done:

1. Neighbour discovery: each node maintains permanently its weight, which is broadcasted in the form of hello message to its neighbours. This hello messages are encrypted using the hello key K_H. During this step every node saves the weight of all its neighbours:

$$E_{K_H^i}(W_i, id_i), MAC_{K_{GI}}(W_i, id_i))$$

$MAC(W_i|id_i)$ is calculated using the hash function.

2. Cluster-head election: According to the weight calculated in the previous step, the node defines its role, to be a cluster head or an ordinary node. The node with the highest weight becomes cluster-head and the others are cluster-members. Every node then broadcast its status encrypted with the hello key:

$$E_{K_H^i}(W_i, id_i, role), MAC_{K_{GI}}(W_i, id_i, role)$$

After cluster-heads are elected, every one between them generates a random K_c and broadcast it to all its cluster members to be used for intra cluster communication.

$$E_{K_H^i}(W_i, K_C^i), MAC_{K_{GI}}(W_i, K_C^i))$$

3. Cluster head backbone creation: After all the cluster-heads have been elected, they must discover each other and setup a cluster-head backbone which later will be used to elect key manager to distribute updated traffic key.

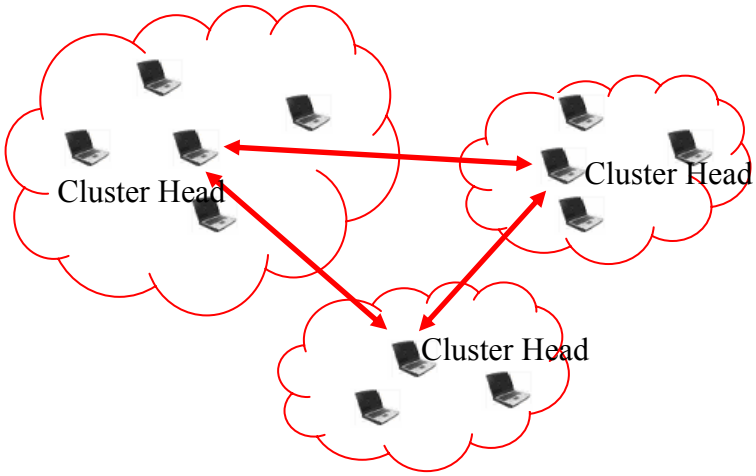


Figure II.8 Clustering architecture

4-1-6 Key update operation

During the key update phase one of the cluster-heads elected in the previous phase will become the key manager and generates the new traffic key that will be distributed among other cluster-heads.

First cluster-heads must collaborate to choose one of them to be key manager. The potential key manager (which has already been a key manager) checks whether any of its neighbouring cluster-heads has a higher weight. The key manager then generates a new traffic encryption key which is afterwards distributed to all other cluster heads over the cluster backbone:

$$E_{K_H^i}(W_c, id_c, K_{TEK}^i), MAC_{K_{GI}}(W_c, id_c, K_{TEK}^i))$$

Then every cluster head distribute this key to all its cluster members:

$$E_{K_H^i}(id_c, K_{TEK}^i), MAC_{K_{GI}}(id_c, K_{TEK}^i))$$

4-1-7 Analysis

- This solution is very suitable for sensor network, where public key cryptography can't be used; therefore it uses symmetric cryptography which is less resources consumption. But the use of symmetric encryption is very vulnerable to long term attacks, where an attacker saves a great amount of data and tries to get the encryption key by using exhaustive attacks, to compromise K_{GI} then he uses the same algorithm used to generate derived keys to generate the keys used in that moment.
- The author proposes to use a key to guaranty authentication, however this isn't sufficient because authentication can't be guarantied using symmetric key encryption. Authentication can only guarantied using public key encryption and certificates.
- An attacker can compromise only one node to gain access to the network.

4-2 Demonstrative Identification

This solution proposed by Balfanz and al in [40] presents a mechanism for bootstrapping trust relationships in local ad hoc networks where the network nodes have no prior relationship with each other. Examples of such local ad hoc networks could be a group of friends wishing to setup a temporary network using their PDAs.

4-2-1 Overview

The solution is based on the connection limited channels; examples of such channels are infrared, physical contact, audio etc. This is justified by the democratisation of the use of these channels, which exist in every mobile device. So the participants in this network establish an initial trust using this connection limited channels, two nodes first exchange authentication data, termed pre-authentication data over a location-limited channel, which is used after to enforce security over the main wireless links.

Figure II.9 Location limited channel

The concept of demonstrative identification can be illustrated with the following example.

Consider a user with a PDA who wishes to connect to one of several printers. By using e.g. an infrared channel the user can identify the printer by going up to it and directing the infrared device towards it. Due to the characteristics of infrared communication the user can be assured that the authentication data exchanged originated from the chosen printer.

Using demonstrative identification and pre-authentication data the authors describe different protocols for two-party and group key-exchange protocols.

4-2-2 Example of use

For two nodes A and B capable of public key cryptography to exchange a secret encryption key the following steps are performed:

1. Using the location-limited channel node A sends $Hash(PK_A)$ to node B and node B sends $Hash(PK_B)$ to node A.
2. Switching to the main wireless channel the nodes now exchange their public keys. The received keys PK_A and PK_B, are verified respectively using $Hash(PK_A)$ *and* $Hash(PK_B)$.
3. If the verification was successful the nodes can exchange a symmetric key used to secure ordinary traffic over the network.

The author in [41] describe a security mechanism for ad hoc network called TAP based on connection limited channels, to exchange pre-authentication with a third party called mediator which is assumed to have more computational power, responsible of the management of security.

4-3 Password Authenticated Key Exchange

This section will describe a key agreement protocols that can be used to secure the communication in a collaborative networking scenario where its members can't use certificate mechanism to ensure authentication due to their heterogeneity, neither symmetric key mechanism or demonstrative authentication described in the previous section.

This solution described by Asokan and Ginzborg in [42] present a password authenticated and group key agreement protocol.

4-3-1 The Hypercube Protocol

The nodes participating in the protocol are arranged as the vertices in a *d*-dimensional cube (hypercube). The protocol then consists of *d* rounds of two-party Diffie-Hellman key exchange. During each round $j = 1,..., d$ a node performs the two-party key exchange with its neighbour. In the first round each node *i* uses its own secret *xi* as the exponent. In the following rounds the key obtained from the previous round is used as the secret exponent to exchange key with other nodes. Figure II.10 illustrates the Hypercube protocol where the number of participants is four, i.e. $d = 2$. In the first round nodes 1 and 2 perform a two-party Diffie-Hellman key exchange; in parallel nodes *3* and *4* do the same. After the first round the pairs *(1, 2)* and *(3, 4)* share a common secret $K_{1,2}$ and $K_{3,4}$. In the second and final round nodes *1* and *3* perform a two-party Diffie-Hellman key exchange. The key exchange in the previous round is now used as the secret exponent in the Diffie-Hellman key exchange. E.g. node *1* sends to node *3* and node *3* sends to node *1*. After the second round is complete all four nodes share a secret K.

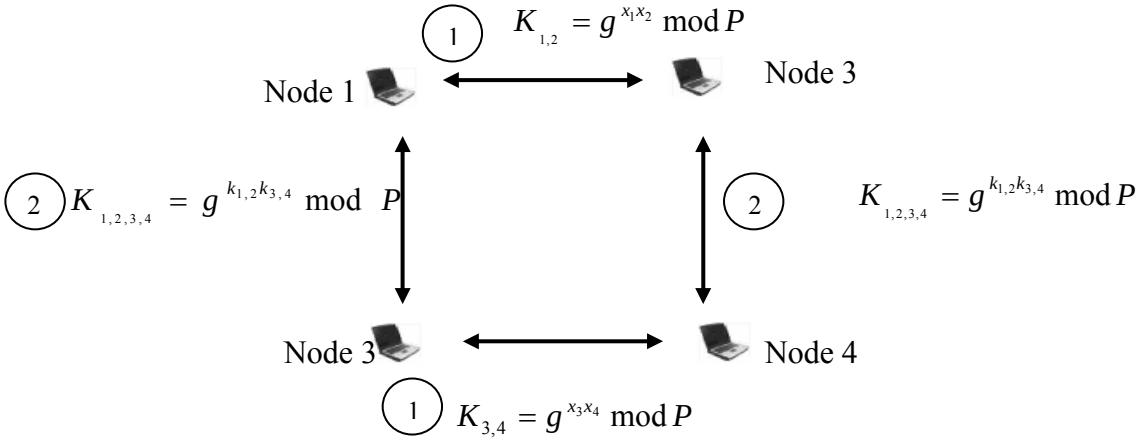


Figure II.10 Hypercube Protocol

The protocol can be generalized to *2d* group members and the members are then arranged in a *d*-dimensional cube and the same steps as described above are performed.

4-3-2 Password Authentication Extension

This mechanism known as Encrypted Key Exchange (EKE) provides a two-party password authenticated using Diffie-Hellman key exchange. The two parties wishing to exchange a strong cryptographic key agree on a simple password *p.* Therefore each party uses this password to encrypt the habitual Diffie-Hellman protocol steps.

This mechanism is used to enforce security and to avoid man in the middle attacks, to which Diffie-Hellman protocol is vulnerable.

1. A→ B: $E_P(g^{x_A} \bmod P)$
2. B→ A: $E_P(g^{x_B} \bmod P)$

After receiving the message in step 1, *B* can extract and calculate the shared key *k* as $K = g^{x_A x_B} \bmod P$ the same thing is done by A when receiving the message from B. in extended situation the password can be written on a board to be used by all the members of the conference.

4-3-3 Analysis

This kind of key management is suitable for heterogeneous network, where there are a limited number of nodes. But in an extended scenario where there are a great number of nodes the speed of the operation may be very slow which affects the bootstrapping of the network. However it can be used in bootstrapping whenever the participating nodes don't have location limited channel to perform the exchange of pre-authentication data as described in the previous section. It can also be used for group authentication to manage the bootstrapping of great number of nodes, by distributing the password to all participants or write it on a board used after to exchange pre-authentication data between this group of nodes.

5 Conclusion

In this chapter we've presented some of most known key management schemes in literature. As we've presented key management schemes are divided into two categories the public key based infrastructure and other key management schemes based on some techniques like location limited channels. As we've presented in each section all the presented key management schemes are vulnerable to some attacks and some of them aren't applicable to ad hoc networks, since they use lot of computational power and messages to ensure their management which isn't desired for ad hoc networks.

We've also observed that some key management schemes like password authentication and demonstrative authentication are suitable for bootstrapping in ad hoc networks

Therefore in the next chapter we are going to present our proposed solution which is based on public key encryption guarantying a robust key management regarding the advantages of this encryption. Our proposed solution is based on clustering to overcome the limitation of effectiveness of the existed key management schemes. We propose also to use password authentication or demonstrative authentication for bootstrapping which are based on human relationship guarantying in this way strong authentication.

Chapter III

Cluster based PKI

This chapter is set apart to the presentation of our proposed key management scheme, since all the schemes presented in the previous chapter suffer from lot of disadvantages making them inapplicable for ad hoc network due to the difficulties of their specifications or they have some security vulnerabilities. The presented solution is based on Public Key Infrastructure using clustering as management technique to simplify the management and distribution of certificates. This chapter is organized in the following sections.

The first three sections give a brief introduction and definitions of our proposed solution, then we'll present some assumption made on the network architecture and the mobile node.

In section four and five we'll present the primitives executed during the lifetime of the network by the network nodes; to ensure its bootstrapping and maintenance including cluster and node management primitives.

Sections six and seven give how the Certificate Authority executes its security and management services using the specifications of our design.

The section eight gives a brief analysis of our design against some known attacks to prove the utility of our design. We also give an analysis regarding the evaluation criteria given in the previous chapter.

The last section of this chapter gives the feature implementation, in which we give a brief description of our implemented prototype. The prototype is implemented in JAVA giving it lot of scale of use. Then we finish the chapter by a conclusion.

1. Introduction

As described before PKI (Public Key Infrastructure) has been recognised as the most efficient and effective tool providing key management in classical networks. However, it is still questionable if PKI can be implemented in ad hoc network or not, considering the characteristics of these networks (bandwidth constraint, dynamic topology…etc). As shown in chapter II different key management schemes have been proposed to implement PKI in ad hoc network, we have also shown that all these solutions don't treat all the key management schemes proprieties as availability, scalability and freshness…etc and other mechanism don't treat the aspect of efficiently, as threshold cryptography solution which include all or a subset of network nodes to provide CA(certificate authority) services, and poses some assumption which are not always guarantied in ad hoc network, like the existence of at least k neighbours in the neighbourhood of each node.

In this chapter, we describe our efforts on providing robust and scalable security services for mobile ad hoc wireless networks. Our proposed design is based on clustering concepts, which consist to elaborate a virtual backbone over the network to ensure CA services. Since clustering is a good alternative to ensure scalability and availability for other services like routing, we think also that clustering may be the best underlying scheme to achieve security services in ad hoc network, if it's accompanied with some security mechanisms, to ensure secure election and backbone elaboration in the network.

In the next chapter we'll present a secured clustering algorithm, trying to include security in the elaboration of the clustering architecture.

Considering that the proposed clustering scheme is secure and efficient. We are going to use it in order to provide the infrastructure for our security scheme, so that some vulnerabilities due the affectation of security services to one node in a given area may be avoided.

Our security scheme is a PKI based scheme, in which we affect the full CA services to one node in a given area this node is the cluster-head. We also use multi-signature certificate, which is a certificate signed by more than one cluster-head, during inter-cluster movement of network nodes.

2. Models for our design

2.1 System and network models

We consider a dynamic ad hoc wireless network with *n* networking hosts/nodes. Nodes communicate with the existence of bandwidth-constrained, error-prone, and insecure wireless channel. Three kinds of network can be considered:

Collaborative networks: is defined as a set of nodes held by humans desiring to elaborate a network to serve their needs. Nodes can be of any kind (laptops, PDA…etc). In these networks we can assume the help of nodes holders to enforce the security mechanisms, making it based on human relationship, to avoid some attacks like impersonate.

Personal networks: consist of a set of handled devices (Phone, PDA,etc) using some location limited channels like Bluetooth, to elaborate a small network serving personal needs of their proprietor. In these networks we can assume the existence of at least one human who is the proprietor of the network, to handle a party of the security management and collaborate in the security of his network.

Sensor networks: these networks consist of a set of sensors which can be deployed anywhere (nature, nuclear area, etc) where the presence of human is difficult or

impossible. Here we can't assume the existence of any human, so the devices must be preconfigured before deploying them.

We assume that nodes use like link layer protocol IEEE 802.11 or IEEE 802.15 presented in chapter I.

We assume also the existence of a routing protocol guarantying the forwarding of packets between nodes which aren't in the transmission range of each other.

Nodes may freely roam in the network. The number of networking nodes *n* may be large and dynamically changing because mobile hosts may join, leave, or fail over time.

We assume that each node is equipped by some limited connection channels as Bluetooth and infra-red, this assumption is easily satisfied since this kind of channels are democratised and they exit in the majority of devices. Otherwise we can use other mechanisms to overcome this limit.

We assume that there is no infrastructure support available, either physical or logical.

We assume also that every node in the network is able to use symmetric and asymmetric encryption. With the possibility of using signature primitives presented in chapter I.

Confidentiality, authentication, integrity, and non-repudiation, are guarantied using both symmetric and asymmetric encryption.

Finally we assume that each node employs some local detection mechanism to monitor its one-hop neighbours behaviour [33], which may be executed in different way according to the degree of importance of the considered node in the network.

2.2 Adversary models

An adversary is a malicious node that uses every available means to break in (such as node compromises) or shut down (such as DoS attacks on servers) the security system.

In our design we use two cryptographic primitives (symmetric and asymmetric primitives). We suppose that the asymmetric cryptographic primitives such as RSA are practically secure in term of the computation power of the adversary. However, the symmetric primitives used to encrypt traffic over the network are vulnerable to long term attacks.

When a networking node is compromised, all its information, public or private, is exposed to the adversary. This information includes the node's public and private key and the network traffic key.

Several adversaries may also conspire into a group to combine their computation power and share their victims. For easy presentation we denote such an adversary group by a single adversary, with extra computation power compared to the rest of the network nodes.

3 System Architecture

Our security design is based on clustering, affecting all the CA services to the cluster-head which ensure intra-cluster security. The inter-cluster security is achieved by the elaboration of a virtual backbone between clusters, ensuring the underlying infrastructure for inter-cluster collaboration to manage security.

3.1 Clustering

Clustering consists of grouping nodes of ad hoc network into clusters (groups), where a node is intended to perform some special tasks like management of security, or routing named cluster-head.

Cluster architecture may simplify the management of ad hoc network, for example clustering for routing reduces significantly the overhead costs imposed by routing. In the sense that every node in the network is identified by its identity and the identity of the cluster to which it belongs. This architecture which may reduce significantly the number of entries in the routing tables is implemented by some protocols like ZRP (*Zone Routing Protocol*) [14], and CBRP (*Cluster Based Routing Protocol*) [15].

Clustering for security management may simplify the key management and ensure some key management proprieties like availability and freshness by maintaining all or a subset of security service on the level of the cluster-head which may be available for each cluster-member at any time, because a permanent connection between cluster-head and cluster-members is assumed by sending beacons. The freshness is ensured because a report of network state is kept on the cluster-head, constructed by collecting event sent by all cluster-members.

Clustering allows a great degree of scalability, since it extensibility is managed by the creation of new clusters, which keeps the same management complexity for any size of the network.

To be used for security purposes we must include the aspect of security when electing cluster-head, which is the case for our proposed clustering algorithm in the next chapter.

3.2 Used clustering algorithm

As we have said our solution is based on clustering. But existed clustering algorithms as Weight Clustering Algorithm [43], highest degree algorithm [44] …etc don't treat the aspect of security when electing cluster-head. And some ones don't pose any restriction on the elected cluster-head to be the most powerful or secured node in the network.

Since the interest of security is a collective objective it must be handled by all nodes in the network. Thus we'll propose in the next chapter a secured algorithm based on voting in order to compute a trust value measuring how much this node is trusted by the others, allowing us to elect the most secure node in the network as cluster-head. We'll also propose a set of criteria that must exist on a node to be cluster-head, concerning its capabilities (Memory, battery…), and its stability. Using such algorithm we can say that the elected node is the most suitable node to be cluster-head.

We think also that the voting mechanism make the cluster-head a subject of observation of all its neighbours, avoiding that cluster-head can be a malicious node.

The cluster-head is in the middle of its neighbourhood (cluster), which make it away from attackers.

Therefore in the remainder of this chapter when referring to cluster algorithm we refer to our proposed clustering algorithm SCA in chapter IV.

3.3 Primitives and Notations

In our architecture, each networking node i is associated with a personal asymmetric pair of keys $< Sk_i, Pk_i >$, any asymmetric algorithm can be used to generate this pair of keys:

Sk_i denotes i's private key.

Pk_i denotes *i*'s public key.

The private key Sk_i is kept secret by node *i*, and it is used to decrypt messages intended to *i* and encrypted by *i's* public key Pk_i, so that it can't be decrypted by any other malicious node. It is also used to encrypt some messages or statements to generate a signature.

To protect the private key Sk_i any mechanism can be used like the protection by password, or by saving it in a smart card, depending on the capabilities of nodes.

The public key Pk_i is published in a certificate and it is assumed to be publicly known to all nodes in the network or at least to its neighbourhood.

We denote $c=(m)_{Pki/\ Ski}$ as the encryption operation of the message m by the public/ private key of node *i*, resulting on an encrypted message c.

We denote sign $(m)_{Ski}$ as the signing process of the message m, using the private Sk_i key of node *i*. The signing procedure is described in the chapter I.

We assume that every node *i* have a certificate $Cert_i$ derived from the x509 certificate and contains other useful information like IP address, MAC address. So that any node can be identified and joined without any ambiguity using its certificate.

Every certificate has two states:

Valid which means that all information contained in the certificate are true.

Revoked state which means that the certificate is invalid, so it must be renewed if it is possible, by contacting the CA.

Every certificate is characterised by an expiration time, after which the certificate is automatically revoked.

Certificate can be verified using the CA public key, which is periodically broadcasted over the network.

Certificate are made public in the network, so that any node can obtain it by requesting the corresponding node, or by consulting the trust authority (CA) which has delivered this certificate, which is assumed to keep a copy of each delivered certificate.

Certification services as mentioned in chapter I including issuing/renewing, revoking, storing and retrieving certificates, are affected to the cluster-head in its cluster. So any cluster-member contacts its cluster-head to get one or all these services.

Each CH maintains three kinds of directories:

- *Cluster-member directory:* contains the certificates of the members of the cluster, it is published for the rest of CMs, so that they can consult it to get or verify any certificate, in that cluster.
- *Cluster-Head directory:* in which it saves certificate of all CHs of the network, it is built when constructing backbone network. And it's maintained whenever new CH is elected. It's also used to verify certificate of strange nodes belonging to other clusters.
- *Cache directory:* the CH saves in this directory the frequently used certificates. Since MANET are characterised by fast topology changing, nodes can go far and returns back in a short time, so keeping their certificate in cache may be useful to avoid additional overhead.

The certificate of the cluster-head is broadcasted over the cluster, allowing cluster-members to be permanently attached to its cluster-head. The certificate is sent in beacons used to manage clusters as it is described in the next chapter.

A backbone between cluster-heads is constructed by exchanging their certificate; it is also used to accomplish some operation like key update, roaming and inter-cluster security services.

Any node in the network can be elected as cluster-head if it has the necessary capabilities and the amount of trust in its neighbourhood.

As we have said in the previous chapter, using asymmetric encryption to encrypt great amount of data is unfeasible, because the process of asymmetric encryption and decryption is very slow, and the result of encryption may be greater than the input message. Therefore asymmetric encryption isn't used to encrypt traffic data over the network; however it's used to sign messages using the digital signature described in the chapter I. So in order to ensure traffic encryption we use symmetric primitives, therefore ordinary message sent over the network is encrypted using symmetric encryption, may be or not accompanied by the signature of the sender. We have choose symmetric encryption to encrypt traffic over the network because it's less resource consumption, and it's very fast when encrypting great amount of data, compared to asymmetric encryption.

We denote K_T the symmetric encryption key used to encrypt ordinary traffic over the network, using any robust symmetric algorithm.

We denote $c=(m)_{KT}$ as the encryption operation of the message m with the symmetric key K_T, resulting on an encrypted message c.

We use proactive key update to update the traffic key K_T after a given period T. Because any existed symmetric algorithm may or will be vulnerable to some attacks, or can be broken when gathering the appropriate amount of data. For example the RC4 algorithm used in the WEP protocol is broken when gathering some millions of packets, which may be very simple in ad hoc network, considering the flooding mechanism widely used in ad hoc network for routing. Therefore the use of proactive key update is unavoidable to overcome this vulnerability. The period T is chosen according to the used algorithm and the length of the encryption key. So it may be short when the key length is short and the algorithm is known as vulnerable as RC4. Otherwise it may be relatively long.

In the rest of this chapter we denote cluster-head as CH, and cluster-member as CM.

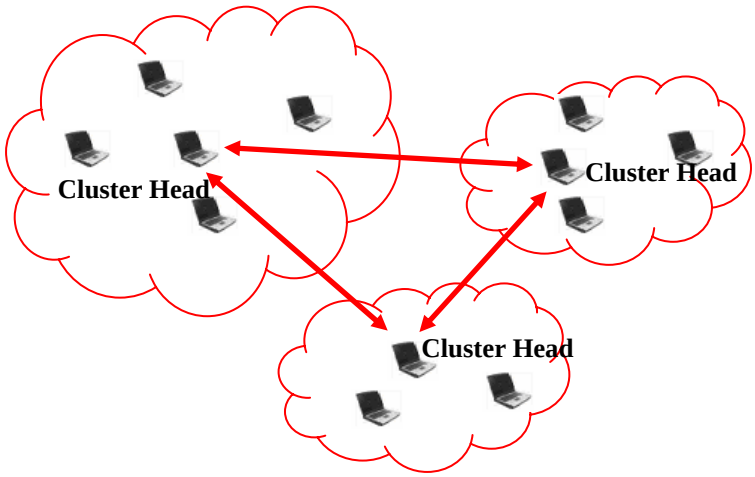


Figure III.1 Clustering architecture

4. System Bootstrapping

We consider a collaborative network, in which there is a set of persons willing to elaborate a network during a conference or for any other purposes. As we have said before, a collaborative network is composed by a set of nodes and their holders, so we can assume that the holder of these mobiles nodes can participate to enforce security in the network. By making the security mechanism based on the human relationship, which is more robust to impersonate attacks.

Thus these nodes must execute collectively the following operations:

1. Since our design is based on clustering, the first step that must be executed is to elaborate the clustering architecture, by executing the election procedure defined in the next chapter. According to number of participant in the elaboration of the network the election procedure can take different forms (we'll discuss this later). We assume also that during the execution of this step all participants are in direct vision of each other, which is evident in this situation, since they can't elaborate a network if they are dispersed in a large area (characteristic collaborative network). After the achievement of the election procedure, the network will be divided on a set of clusters; the number of clusters depends on the capabilities and the transmission range of the CHs. After this the CHs collaborate between them to construct the backbone used for inter-cluster communication. Then each node registers with the nearest CH, by sending its certificate to the CHs for being signed, which means that this node becomes a CM of the cluster managed by this CH.
2. Both CHs and CMs execute cluster maintenance primitives defined later, like roaming and cluster merging. They must also collaborate to accomplish some additional services like revocation, and registration.

4.1 The elaboration of cluster architecture

In this section we are going to explain the process of elaboration of the cluster architecture, using the algorithm described in the next chapter SCA, however we must add some additional criteria to make it robust and solid against some attacks :

The first attack is impersonating attack, in which a malicious node performs a denial of service attack against an honest node in the network to block it and exclude it from the election procedure. Then this malicious node spoofs the identity of the honest node and participates in the election procedure as a legitimate node, and then it gets the symmetric key and registers itself with the network and becomes member of that network.

Another kind of attacks can be executed is passive attacks, to get some confidential information about the system and participants, by eavesdropping the exchanged data during the bootstrapping. Information about topology may be used to perform denial of service attacks, or to infiltrate into the network using any unpredictable attack, as well as private information (name, mail…etc) which can be used after in order to divine passwords used by the same person in other domains using social attacks.

To avoid all these attacks we propose to elaborate a secure area before launching the election procedure.

4.1.1 Secure area elaboration

As we have said during the bootstrapping of the network several confidential information can be eavesdropped and used after to perform attacks against the members of the network as well as the security of the network. So we propose first to elaborate a secure area before exchanging any important data, by constructing a wall

between the participant in the bootstrapping of the network and other exterior persons. We define two techniques that may be used to elaborate this secure area.

4.1.1.1 Demonstrative authentication

This solution described in the previous chapter, is proposed by Balfanz [40], is based on the use of location limited channel like *Bluetooth* and *infra red*. The users use these channels to exchange confidential data, so it can't be eavesdropped because the transmission range of these channels doesn't exceed 10 meters, which means that any attacker is detected by the users. This channel is used to exchange pre-authentication data used after to elaborate a strong channel over wireless link.

The users can use over these channels any other mechanism to enforce security like deffie-helman protocol or TLS (Transport layer security) [19], which are sensible to man in the middle attacks, which can't be executed with location limited channels.

Whenever the secure channel is elaborated the users negotiate the parameters to elaborate a secure connection over the wireless connection using a symmetric key or any other mechanism supported by the users. Whenever the parameters are established we can say that the secure area is elaborated and we can proceed to the election procedure.

This process is used when the number of nodes is small, because they can be near to each other in order to use location limited channels, but whenever the number of nodes exceeds 10 or 20 nodes, it's impossible to make contact with all the nodes. Thus we must use password authentication.

4.1.1.2 Password authentication

Whenever the number of nodes in the bootstrapping exceeds the threshold of 20 nodes, we propose to use password authentication. As described in chapter II this mechanism is based on a password publicly known in the network used to elaborate a secure area, using for example a symmetric key derived from this password, or used to secure a channel between the user and the node chosen as server responsible of the management of the security among the participant using deffie-helman. This server is responsible of the distribution of the symmetric key used to secure the area, thus every node gets the symmetric key used to elaborate the secure area from this server.

4.2 Election procedure

Whenever the secure area is elaborated, nodes can use the SCA algorithm presented in the next chapter to elect the appropriate CH, intended to manage the security over the network. In SCA we have included the aspect of security when electing CH, using a trust value measuring how much the network members trust this node.

We suppose that only some nodes in a given area desire to be CH, so they first send within the neighborhood of D hops a CH_Ready message containing their certificate.

Nodes receiving this request, estimate the appropriate value affected to this node and send it to the requesting node.

After a period T, each node calculates its weight. This weight is broadcast to the neighbourhood then the node with the maximum weight is elected as CH.

After the achievement of this operation, each node is CH or an ordinary node.

4.3 services launching

Whenever the election procedure is finished, we assume the existence of at least one CH in each neighborhood of D hops. These CHs are ready to elaborate the virtual backbone between CH, to accomplish inter-cluster service. Therefore each CH broadcast over the entire network a request containing its certificate which is assumed to be signed with its private key intended to all CHs in the network, *{CH_Creation ($Cert_{Chi}$) }*. Whenever this request is received by any CH it saves it in the cluster-heads directory, and sends him a response containing its certificate, the response is directly forwarded to the CH because its IP address is included in the certificate. This operation is closed after a given period, after which we assume that a backbone enclosing all CH is constructed.

At the same time of the elaboration of the backbone network between clusters, the CH sends beacons within the area of D hops.

We keep the same structure of beacons as defined in SCA, using as identity the certificate of the CH. The beacons sent are ordinary beacons to allow network extensibility.

Every node when receiving these beacons choose the nearest CH, if it hears more than one cluster, and send him a join request containing it certificate for being signed.

The CH signs every received certificate with its private key and stores it in the cluster-member directory. It also sends the signed certificate to its owner.

After a period of time T, one of the CHs generates a random symmetric key and sends it to CHs encrypted with their public key. It also does the same thing with all its CMs. Whenever this operation is achieved all network members use the new symmetric key to encrypt ordinary traffic over the network. This symmetric key is then updated after a given period.

5- Network maintenance

Whenever the previous step is successfully finished, both CHs and CMs collaborate to maintain the structure of the network stable as possible, by executing the appropriate maintenance operations in response of some event occurring along the lifetime of the network. These operations are the same defined by the underlying clustering protocol. However due to security needs some parameters are changed.

We can divide the maintenance operations into two categories, those trying to maintain the cluster architecture called Cluster Management Operations, and those managing nodes movement (roaming, joining…etc) called Node Management Operations.

5-1 Clusters management

The cluster management operations, try to keep as possible the stability of the network, by keeping the same CH in its state as long as possible, and manage the extensibility by creating new clusters.

5-1-1 Cluster Creation

The cluster creation operations try to handle the extensibility of the network by creating new clusters. In the previous section we have presented the procedure of bootstrapping which deals with the same problem (the creation of clusters). However this amount of clusters may be insufficient if the number of nodes in the network increases, or too small if the number of nodes decreases. To manage the need of extensibility we define the following operation:

5-1-1-1 Clusters birth

One of the primordial characteristics of ad hoc network is the dynamic topology, caused by the movement of the network nodes, therefore any node moves around the network to go far from its CH. Whenever a node detects that it is at *n* hops (*n* is a predefined value, which is greater than the radius *D*), it executes one of the following operations:

1- First it looks for any other CH by hearing beacons, if there is any CH in its neighbourhood, then the node execute the roaming operation, to register with this CH and becomes member of that cluster. Otherwise it tries to execute operation 2.
2- Whenever there is no CH in the neighbourhood, so it is necessary to launch the electing procedure:
    a. It broadcast a CH_Ready beacon, informing its neighbours about the need to create a new cluster.
    b. Nodes receiving this beacons response by sending the trust value estimated for this node. Otherwise they send a CH_Ready beacon if they want to be CH.
    c. The operation of election is achieved by electing one node as CH. This may be other than the node which has launched the operation.
3- If there is no response when sending the CH_Ready beacons. Thus the node declares itself CH, and creates its own cluster.

Whenever one of the operations 2 or 3 is executed, the result is a CH, which must continue the process by executing the following operation with the collaboration of other CHs to update the topology of the cluster heads backbone:

1- It signs its certificate with its private key, and keeps the signature of the old CH in the same certificate giving it more credibility because it's signed by more than one CH. We suppose that in certificate structure there is a mechanism to keep more than one signature in the same certificate (using arrays for example).
2- It sends to its old CH a cluster-head creation request containing its certificate *{CH_Creation ($Cert_N$)}*, informing him about the creation of a new cluster.
3- When receiving this request the CH verifies its validity and responses by sending the list of all the CHs in the network.
4- The new elected CH sends a cluster creation message *{CH_Creation ($Cert_N$)}* to all CH contained in the list sent by its old CH.
5- Every CH when receiving this request verifies the validity of the old signature and stores the new certificate in the CH directory.
6- After receiving confirmation from all the CHs, the new CH begins sending ordinary beacons, to allow the extensibility of its cluster.

5-1-1-2 Cluster division

This operation is defined to handle the aspect of resources limitation of the nodes in ad hoc networks. Since these nodes have limited memory and battery, so they can't serve a great number of nodes for long time, and our purpose in a cluster architecture is to maintain a CH in its state as long as possible. Thus we have proposed in the next chapter to use an upper bound of the supported node by any CH according to its capabilities regarding its remainder of battery, memory and transmission range.

Whenever the number of CM in a given cluster increases and reach a certain threshold, and there are new joining requests, or when the CH is busy, it must execute the following operations:

First the CH tries to solve the problem without the creation of a new cluster, by sending in its beacons a roaming command. Whenever this command is received from CMs, every one tries to roam to the nearest CH. After receiving the appropriate confirmations from nodes which have roamed to other clusters, the CH continues its habitual operations. Otherwise it must execute the cluster division operation as follow:

1- The CH launches in its cluster an election procedure, by sending in its beacons a cluster division command.
2- Whenever these beacons are received by the CMs, the ones between them willing to be CH launch the election procedure as described above. And compute their weights.
3- Each CM sends its weight to the CH.
4- The CH chooses as CH the node having the greatest weight and the most far from the center of the old cluster, to avoid that two clusters be neighbors of each other.
5- The new CH continues by executing the backbone update operation to maintain the structure of the network, and begins sending beacons to allow neighbor nodes to roam to its cluster.
6- The old CH sends in its beacons a roaming command, telling its CMs to roam to the new created cluster.

5-1-3 Cluster merging

The merging process consists of merging adjacent clusters with little number of CMs in a great cluster with reasonable number of nodes, without exceeding the maximum of nodes supported by new CH. The aspect of merging clusters aims to avoid complexity due to manage lot of clusters. Because any CH has to exchange with all CHs, great amount of data to manage security and cluster architecture. The second aspect treated when merging clusters is the security because the failure point of cluster architecture is the CH, so increasing the number of CH means increasing the risk of attacks.

This operation is launched whenever the number of CMs in a given cluster becomes less than a predefined value *Min*. *Min* is the same for the entire network. Therefore whenever the number of CMs decreases to reach the threshold of *Min,* the CH executes the following operations:

1- The CH looks for the nearest CHs, if there is any one then it continues the rest of operations, otherwise it aborts the operation.
2- It sends to its CMs a demission request, telling them that they must leave the cluster, and join the nearest CH, by executing the roaming operation.
3- The CH stays until receiving confirmations from its entire CMs after roaming, and then it roams to the nearest cluster.

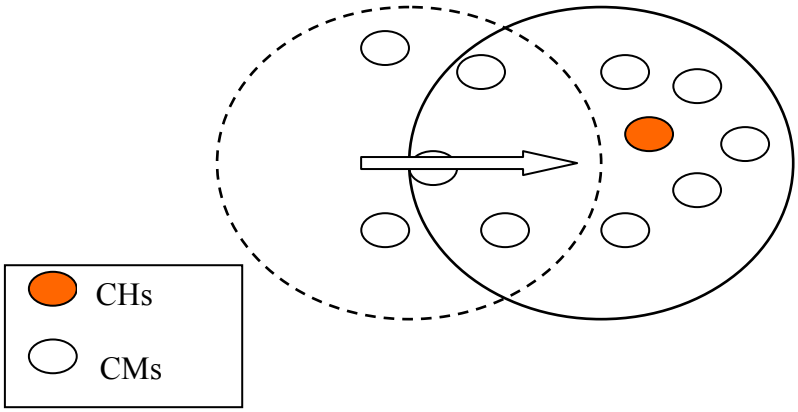


Figure III.2 Cluster Division

5-2 Node management

5-2-1 Joining the network

In chapter II we have spoken about the self issued scheme or PGP, in which every person issues her certificate and signs certificate for the others, this solution is based on human contact, because every person signs the certificate of another if she knows him well.

In our scheme we use the same idea to identify new coming persons. Thus any person can't join the network if she wasn't identified by at least one person, to include him into the network. Because the network is protected by a symmetric key encryption which makes the joining process impossible without a gateway. This gateway plays the role of an intermediate between the CH and the exterior node (person). In the sense that the intermediate node takes the certificate of the new coming person who is assumed to be honest in this situation, otherwise he can't find any person to include him in the network. This is done in the same way as PGP but in our solution the list of certificate is kept by the CH and not by each member, making it available at any time by contacting the CH. In this way we overcome the disadvantage of the self issued certificate which is the limitation of its effectiveness during the initial phase.

Another point is when sending the certificate for being signed over the gateway, in this moment if we use the same idea as described in PGP the certificate of the new node can be stolen making private information like name, IP and MAC address available to attackers which can be used to perform social attacks against this person. So in our solution we propose to use location limited channel to perform the initial phase. Location limited channels characterised by a limited transmission range may limit the risk of stealing the certificate.

Whenever a new person wants to join the network she executes the following operations:

1- The holder of the new node looks for a person to merge him in the network. This person isn't always the holder of CH node; we call this node Gateway GW.
2- The new node elaborates a secure channel with the GW by using location limited channel or password authentication to achieve the remainder of operations.
3- If it can't generate the pair of Public/Private key, it must choose a trusted person (friend…) who has this capability to accomplish the key generation for him.

4- The new node creates a certificate containing several information (Public key, IP and Mac address etc), and sends it to the CH to be signed over the GW.
5- The CH verifies if this information haven't been revoked before. This operation is done by contacting all the CHs of the network.
6- The CH signs the certificate and sends it to the new node over the GW.
7- Using the Public key of the new node the CH encrypts the symmetric key of the network and sends it to that new node over wireless connection.
8- The new node is now member of the network and can benefits from services of the network.

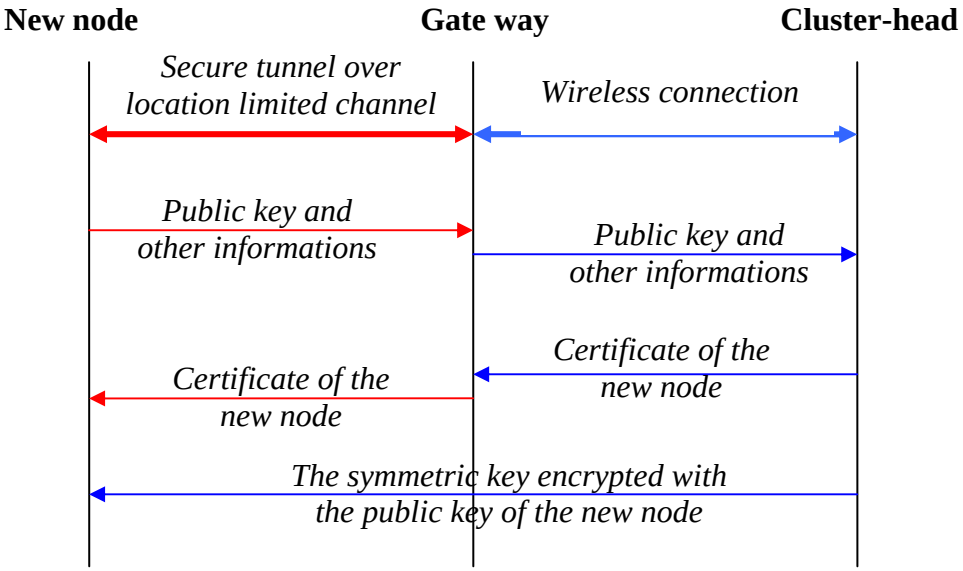


Figure III.3 - Authentication Protocol of Cluster joining process

5-2-2 Roaming

The operation of roaming consists of changing the cluster of a given CM, by detaching it from the old CH and attaches it to the new CH. The concept of roaming is used in lot of domains like GSM, for maintaining permanent connection between the system and its component like phones in GSM. In addition of traditional use of roaming, we use roaming to handle other event in the network, like cluster division, cluster merging and link failure between CM and CH.

In ordinary situation the roaming operation is launched whenever the node detects that it is at more than D hops from its CH, and there is at least one CH in its neighbourhood, therefore it launches the roaming operation by executing the following operations:

1- The node sends to the desired CH a roaming request *{roam_req(Certi)}* including its certificate which is signed by the old CH.
2- The CH verifies the validity of this certificate.
3- If the verification fails an alert is launched to exclude this node from the network. Otherwise a delete request *{delete_req(Certi)}* is sent to the old CH to remove this node from his directory. The old CH moves this certificate from the certificate directory to the cache directory for future needs.
4- The old CH sends a delete reply *{delete_repl()}* to the new CH.

5- The new CH signs the certificate of the new node with its private key to achieve the roaming process.
6- The signature of the old CH is always kept in the certificate (concept of multi-signature).

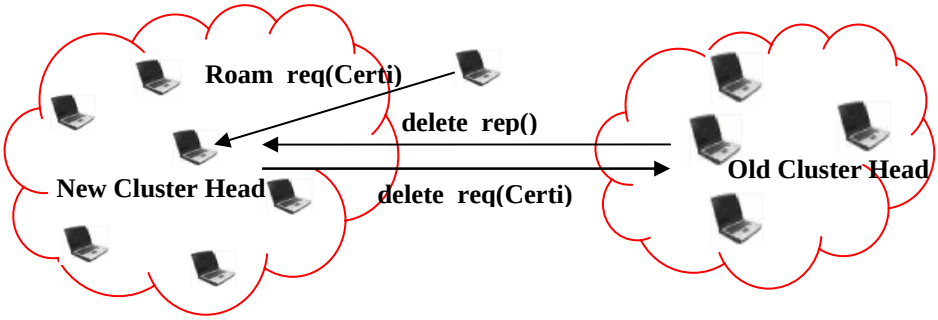


Figure III.4 Roaming process

5-2-3 Leaving the network

As we've presented above every new coming node registers with a CH and becomes a member of that cluster, it can also roams from one cluster to another during the lifetime of the network, until it leaves the network. Thus we define the leaving process allowing a node to declare itself out of the network and liberate the resources (Memory space, emplacement in CM directory…etc) allocated for this CM on the system. Two kinds of leaving can be defined:

5-2-3-1 explicit leaving

The process of explicit leaving is launched by the CM when it decides to leave the network. Hence CM *i* willing to leave the network, sends a leaving request {*leave* ($Cert_i$)} containing its certificate to the CH to which it belong. When receiving this request the CH deletes the corresponding certificate, as a result the CM is declared out of the network, and it's not included in any future operation like key update. Whenever the node needs to join the network it must executes the join process described above.

5-2-3-2 Implicit leaving

The operation of implicit leaving is launched by the CH, whenever a given CM doesn't confirm its dependency to the network for a certain period, this is detected when executing key update operation, therefore the CH contact other CHs to verify if this node is in another cluster. If the node isn't in any cluster it's out of the network and its certificate is deleted to liberate resources allocated to this CM.

5-2-4 Link failure

In this case we suppose a link failure between the CH and all or one of its CMs. Therefore any node detecting this link failure launches a roaming request to join the nearest CH, if there is any CH in that neighbourhood. Otherwise it launches an election procedure to elect a CH and manage the situation.

Whenever the CH is lost (a system fail or it leaves the network), launching the election procedure is very suitable, to delegate another CH which may be fastest than the roaming of each CM.

6- Certificate authority services

In this section we are going to present how our design executes the habitual CA services ranging from revocation, publishing and renewal. As it is known our design is based on affecting the full CA to the CH.

6-1 Certificate structure

As mentioned above our certificate structure is inherited from the X509 certificate Figure 4, however this one doesn't include the aspect of mobility and networks in its structure. Therefore in our design we add some fields to adapt X509 standard to our scheme:

- The IP address: it gives more dependency between the certificate and network node, it also gives the possibility to identify the CM by using its IP included certificate,
- The MAC address: used to make dependency between the physical address and the certificate.
- An array to allow multi-signature in which we save the signatures of all CHs having signed this certificate during the roaming between clusters. This occurs whenever a CM roams to a new cluster, therefore the CH of that cluster signs the certificate and save the old signature in the same certificate, this gives more credibility to this CM when roaming to other clusters.
- Another field is used to save the information about the CH. It's a record containing the Name, IP and MAC address of the CH.

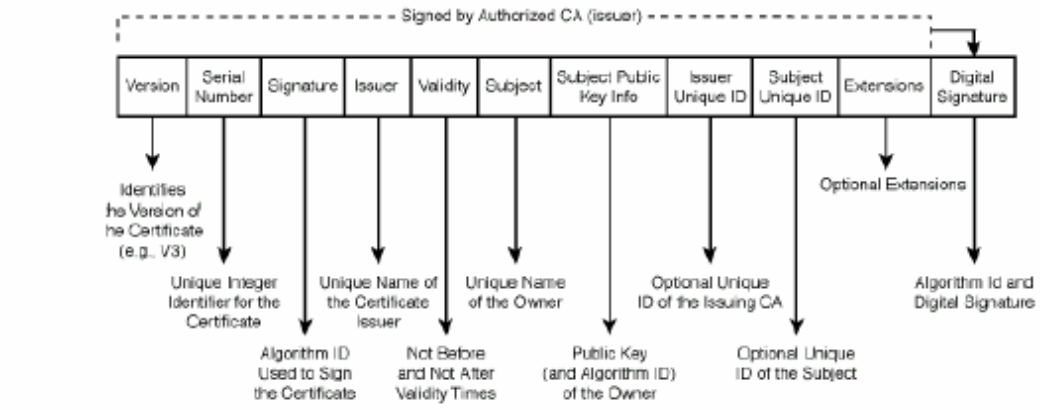


Figure III.5 X.509 v3 Certificate Structure

6-2 registration

The registration process allows a new coming node to register with the CA of the network, as described above the registration process is achieved by executing the joining process explained above.

6-4 Certificate Revocation and CRL

As described in the first chapter the revocation mechanism in PKI is used to exclude a certificate from the system. This may be caused by the compromising of certificate because it is used by a malicious person or the expiration of the validity period of this certificate. In our design we define two kinds of revocations:

6-4-1 Explicit revocation

This mechanism is launched when a malicious node using a legitimate certificate is detected, so it must be excluded from the network. This is done by moving its certificate to the revocation list, and sends an alert over the entire network to exclude it immediately from the network.

Hence when a revocation process is launched the corresponding CH diffuses the revoked certificate to all CHs. Then it launches the key update mechanism described in section 7-2. The holder of the revoked certificate isn't included in the new key update, which means that it doesn't receive the new symmetric key making it out of the network.

Our proposed scheme for revocation is the same as the one proposed in [33], based on accusations which are sent to the CH. So any node must observe the behaviour of its neighbours and sends accusations whenever a doubt on the behaviour of a node is detected. When the number of revocation reaches a certain predefined value, its certificate is automatically revoked. Therefore we suppose that every node implement a subset of an Intrusion Detection System IDS [45].

6-4-2 Implicit revocation

Every certificate when is generated has a validity period, after which it becomes invalid. This mechanism is very useful to give for some persons a limited period to use the network services. In this situation when a CH detects that a certificate becomes invalid, it sends a revocation request to all CHs to exclude this node from the network. Then it launches a key update operation to change the traffic key and consequently the node becomes out of the network.

6-5 renewal

The renew mechanism in PKI allows a node to change any information in its certificate. Because the information in the certificate are valid only for a certain period (Mail, IP and MAC address…etc), or whenever the certificate's validity period is expired. In order to accomplish the renewal process the node executes the following operations:

- The node *i* sends to the CH a renewal request message {*renew* (Old_Cert$_i$, *New*_Cert$_i$)}, including the old certificate, and the modified certificate.
- The CH verifies the validity of the old certificate and signs the new one.
- The CH stores the new certificate in his directory, sends a copy to the node, and deletes the old certificate.

7- Security services

7-1 Identification of nodes

The identification process is launched when two strange nodes become neighbours; therefore each one must identify the other.

When a node A needs to identify a neighbour node B, it asks him for its certificate, therefore A sends a verification request to the CH. The CH who has the certificates of all CHs verifies the validity of the certificate and sends a response to A. If the response is positive then B is authenticated. If the response is negative; because its certificate is revoked, then B is a malicious node. So A sends an alert message to the CH who launches the revocation process to exclude this node from the network. The same operations are done by B to authenticate A.

7-2 Key update

As we've mentioned the mechanism used to encrypt traffic is based on symmetric encryption because this one is less resources consumption compared to public key encryption. In the literature there are several symmetric algorithms (RC4, DES, AES…), but the ones used are not very robust. To avoid traffic key compromising we propose the use of a proactive key update to periodically change the traffic key. So one of CHs generates a new traffic key K_T and an appropriate timer allowing the synchronization of the use of the new key, and then it encrypts them and sends a copy for each CM encrypted by the public key of each CM. It also does the same thing with all CHs. When receiving the new key every CH encrypts a copy with a new timer for each CM by using the public key of that CM, when receiving the key every node sends a confirmation message to its CH. The network nodes wait until the timer expires to launch the use of the new key. We've supposed that the CHs are synchronized during the virtual backbone elaboration.

The mechanism of key update can be used to exclude node from the network as described above, whenever it certificate is revoked, this is done by excluding this node from the key update operation, because if a node doesn't receive the new traffic it's considered out of the network.

8- Analysis

In this section we are going to give a brief analysis of our design regarding its resistance to attacks and the accomplishment of the key management characteristics. We'll also compare our scheme to other existed schemes taken from the literature:

8-1 Resistance to attacks

In this paragraph, we try to evaluate the robustness of our solution regarding to security threats. By the way, we will focus on different main attacks:

- *Listening to packets:* this passive attack could be easily solved by using an encryption technique. In our approach, we use symmetric key encryption with the mechanism of proactive key update which may enforce security, constructing in this way a wall between authorized member of the network and exterior nodes trying to eavesdrop the network traffic.
- *Impersonation:* in this attack the attacker uses the identity of an authorized node in order to get access to network resources. In our scheme we use certificate which are very robust against impersonate attacks, in addition we've based the authentication process on human relationship, which is robust to impersonation attacks, because

only honest persons are authorized to join the network. Therefore each member of the network is responsible of the security of the network.

- *Modification:* to protect data from modification we use both symmetric and asymmetric key encryption. In the way that we use asymmetric key encryption and digital signatures to provide data integrity, and symmetric encryption to ensure privacy.
- *Insertion:* in this attack, the attacker inserts data pretending that they have been sent by a legitimate node. This attack can be easily solved since our approach ensures robust authentication. Thus, only authenticated nodes can inject useful data in the network and even if attackers succeed injecting data in the network this data will be rejected because it's not registered with the network, this may occur if he gets traffic key, which isn't possible without the registration with the network, otherwise he can't inject any data
- *Denial of service:* here, the attacker limits or blocks several network services. The attack can't be executed by exterior nodes because he can't get access to the network since he isn't authenticated. However denial of service attacks targeting the physical layer (interference) are always possible.
- *Routing attacks:* these attacks target the routing protocols by modifying routing information or by injecting false information. These attacks aren't possible to be executed since the traffic including routing information is encrypted, in addition we can use any other mechanism taken from the literature to improve the security of routing protocol, using digital signature to ensure authentication and integrity of the routing information.

8-2 Key management characteristics

- *Fault Tolerance:* This criterion is guaranteed because the certificates signed by a given CH are valid along the life time of the network. So the loss of the CH doesn't affect the validity of the certificate of any node and every node losing contact with its CH may roam to another cluster using the same certificate.
- *Security:* the security in our framework is guaranteed by the use of two mechanisms. The first one is the use of the symmetric encryption which provides confidentiality. The second one is the use of public key encryption which guaranties integrity, non repudiation, and authentication.
- *Availability:* our approach is a cluster based architecture, where every CH is CA of its cluster. This aspect guaranties availability because the CH is always in neighbour of its cluster members. And when it is not available the members can change the cluster without the loss of their certificate which may be valid along the network lifetime.
- *Freshness:* this aspect is established by the key update mechanism, and renewal of certificate.
- *Scalability:* this propriety which is dealing with network widening is possible in our approach because when the network increases the number of cluster increase as well, by electing new CHs and creating new clusters which simplify the management of the network. This means that the complexity remains the same.

8-3 Comparison with other PKI key management schemes

As we've described in chapter II lot of PKI management schemes exist in the literature, however every solution has some limitations (Chapter II), to overcome the limits of these solutions we've proposed a cluster based PKI in which the CH plays

the role of the CA ensuring in this way lot of key management criteria like we've see in the previous section.

Compared to fully or partially distributed solution our solution is efficient since these two solutions suffer from efficiently when the number of nodes get higher, however our solution uses clustering to manage the increasing number of node, and it doesn't use any complex computing to deliver key like fully or partially distributed which uses shamir's secret sharing to distribute the key among nodes in the network which is too difficult and complex which may not be supported by mobile nodes characterised by limitation of their computational power.

Compared to PGP which is suitable for ad hoc network but it effectiveness is limited during the bootstrapping, our solution ensures the effectiveness during all the lifetime of the network always by using clustering.

The overhead is less and constant compared to other schemes needing more messages to execute CA services.

It allows a mutual and strong authentication between CHs and new joining nodes, using human relationship, trusted gateway and location limited channels.

It uses symmetric encryption to encrypt ordinary traffic over the network which is more suitable and less computational consumption making it simply adaptable for sensor networks.

It uses proactive key update to enforce security and to revoke malicious nodes.

9- Implementation

In this section we are going to describe the implementation of the prototype of our protocol design using Java language [46]. We have used Java to give more portability and modularity to our protocol design, so we can use different platforms to test it. To allow communication possibilities over the network we have used the RMI (Remote Method Invocation) mechanism [47].

The RMI is a mechanism which is used to develop distributed applications over networks using only JAVA language. Our choice to uses RMI mechanism is justified by the simplicity and the transparency of transferring of JAVA object from one host to another. Therefore object like certificate, beacons or the state of nodes in the network can be easily exchanged.

9-1 Class diagram

The prototype is mainly composed of the three following classes:

- *Cluster-head:* this class implements all the operations that must be executed by a CA (registration, revocation, renewal…etc), it also implements some data structure to save CHs and CMs certificate like the revocation list and CHs, CMs and cache directories (Figure III.5). To be executed for test we've implements other methods and data structure to manage the mobility and the specificities of ad hoc networks like the coordinate and speed of the CH. We've also implements methods to manage beacons sending, nodes registration and certificate renewal and revocation.
- *Cluster-member:* this class implements the structure of any node in the network, therefore it redefines the data structure of the CH like (coordinate, speed, name…etc). It implements all operations which must be executed by any CM when joining the network in collaboration with the CH, like revoke, register…etc (Figure III.5).
- *Certificate:* this class implements the structure of a certificate as defined in section 6.1. In our design we have used a subset of X.509 standard and we have added other information like mac-adress, ip-adress… (Figure III.5).

- *Beacon:* this class defines the structure of the beacon messages sent by CH to CMs (Figure III.5). It contains the certificate of the CH.

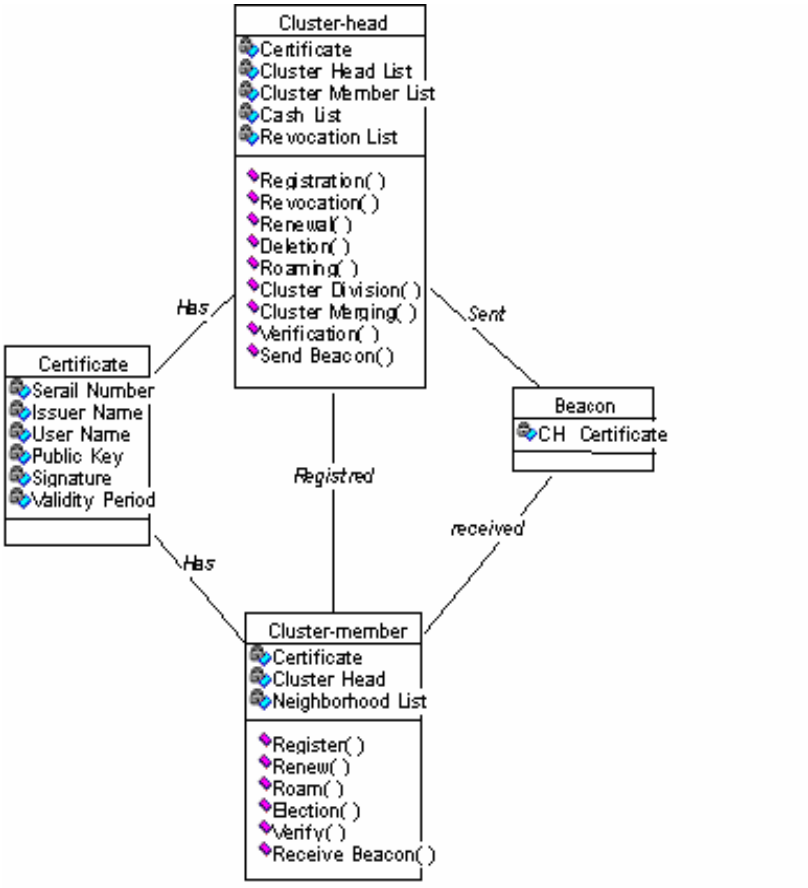


Figure III.6 Simplified class diagram

A CH can serve simultaneously lot of CMs, in order to allow parallelism we've used Thread [46]. So any operation like registration, renewal, roaming and beaconing …etc is handled by a thread which allows that the service will be available at any time for each CMs.

9-2 Description of some process

In order to simplify the description of the prototype we can divide the operations executed in two parts, those executed in background and those needing the collaboration of the node holders (user handled operations):

9-2-1 User handled operations

The CM side of the prototype is presented as a user interface with a main menu allowing the user to execute the following operations

9-2-1-1 Key generation

This operation is executed from the main menu allowing the CM to generate an asymmetric pair of keys < Sk_i, Pk_i > using java primitives defined in the package java.security.*.

9-2-1-2 Registration

This operation combines the public key and other information needed for the creation of the certificate (section 6.1), and sends then to the CH to create and sign a certificate. If the user is authenticated then the CH signs the certificate and resends it back to this node, as a result it becomes member of that cluster.

9-2-1-3 Revocation

This operation consists of sending an accusation request against a node in the network. Whenever the user doubt in the behaviour of any node he send an accusation to the CH, when the number of accusation reaches a certain threshold the certificate is automatically revoked (section 6.4).

9-2-1-4 Certificate Renewal

To execute this operation the user must first change his personal (including the pair of keys) data, then he sends the new created certificate accompanied with the old one to the CH to be signed.

9-2-1-5 Leaving the network

This operation is executed from the main menu File and it's intended to detach the node from the network liberate the memory space allocated for this node.

9-2-1-6 Getting information about the network

As seen in Figure III.6 this command is presented on a panel allowing a node to observe the state of the network at any time including the existed CHs and their CMs. Other information concerning the encryption algorithms are also available. This information is periodically obtained from the CH.

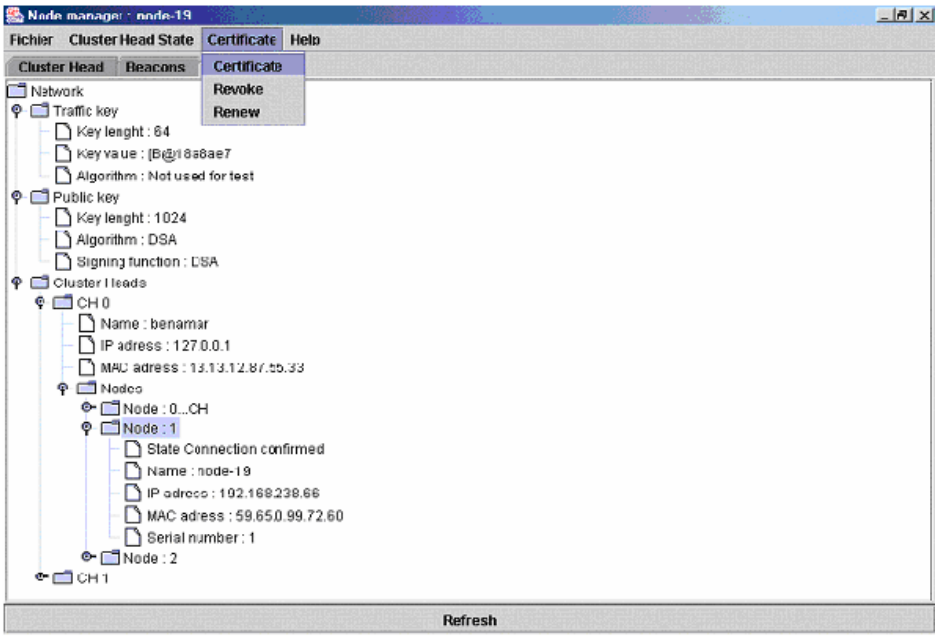


Figure III.7 prototype user interface

9-2-2 Background operations

These operations are executed in the background, which means that it doesn't need the intervention of the user. Each operation is managed by a thread on the CH, and it has two implementations according to the node executing this operation (CH or CM).

9-2-2-1 Beaconing

The operation of beaconing when executed by the CH consists of generating a beacon containing the certificate of the CH and its new coordinate. However when executed by the CMs it's intended to get periodically the beacons sent by the CH and decides the next operation to be executed (roaming…etc)

9-2-2-2 Getting traffic key

This operation is periodically executed by CMs to get the new traffic key, randomly generated by the CH. We have used as random number generator the java class java.util.Random.

9-2-2-3 Moving in the area of the network

This operation is executed in the same way on both the CMs and CHs. It's managed by a Thread which is executed each second to change the coordinates of the network according to the speed of the network which is randomly affected to each node during the initialisation phase.

9-2-2-4 Roaming

This operation is launched by a node whenever it detects that it's far from its original CH. It consists of detach the CM from the original CH and attach it to the new CH.

9-3 feature for real implementation

In the above section we've presented only a prototype to test the performance and the feasibility of our protocol design, however for being implemented in reality we gives an implemented for each kind of mobile nodes using different version of JAVA editions.

1. Using the Java standard edition J2SE [46] intended to be executed on laptop with great resources.
2. Using Java micro edition J2ME [48] intended to be executed on handled device with limited resources like PDAs, Mobile phone.

However when implemented using J2ME we must take into consideration the limitation of this edition. Because the specification of J2ME doesn't include the aspect of generating asymmetric keys, and perform public key encryption and decryption.

10- Conclusion

In this chapter we have presented our key management scheme, which is based on the concept of clustering to simplify the management of the network, by dividing it on a set of clusters (groups). In our scheme we have chosen the CH as the CA of its cluster to be available at any time. We have also defined other mechanisms to manage mobility like roaming, multi-signature and other mechanism to manage the bootstrapping and nodes registration making them more robust and secure. We have also proposed to use SCA as clustering algorithm, since SCA include the aspect of

security and other mechanism making it more suitable for security purposes. We have shown that our scheme is very robust against several attacks since it ensures confidentiality using symmetric encryption to encrypt traffic, as well as authentication and integrity using asymmetric encryption. We assume also that our scheme is working independently of any routing protocol, which give it the possibility to be used over any existing routing protocol. In the last part of this chapter we have presented a prototype to describe our scheme which is written in Java to give it a large scale area for test.

Chapter IV

Secured Clustering Algorithm

In this chapter we are going to present our proposed scheme of clustering. In the previous chapter we've presented a key management scheme based on clustering, however the used clustering schemes don't treat the aspect of security. Therefore in this chapter we'll present our contribution in which we try to overcome some limitations in existed algorithms. This chapter is organized into four sections.

In the first section we give the definition of clustering its advantages and area of use, we'll also give a set of criteria which must be guarantied by any clustering algorithm to be used for ad hoc networks.

The second section is a state of the art of clustering algorithm, five algorithms are presented each one is followed by an analysis section in which we treat its limitations and advantages.

The third section is consecrated to the presentation of our clustering algorithm and the ideas behind it. Our clustering algorithm is called Secured Clustering Algorithm, since it include the aspect of security in the election procedure, it also uses a set of innovative ideas as stability computing and the use of a set of system parameters making its possible to be used for different configuration of ad hoc networks.

The fourth section give the interpretation of the experiment results of the tests made on Secured Clustering Algorithm, we also give the results of the comparison of our algorithm with other algorithms taken from literature.

1-Clustering

Clustering consists of grouping nodes of ad hoc network into clusters (groups). Where one of theme is intended to perform the management of the cluster named cluster-head. Cluster-heads are responsible for the formation of clusters each consisting of a number of nodes (analogous to cells in a cellular network) and maintenance of the topology of the network. The set of cluster-head is known as a *dominant set [40]*. The cluster-head allocates resource for all the nodes belonging to its cluster, to accomplish needs of the underlying protocol using this clustering algorithm (routing or security).

Any node can become a cluster-head if it has the necessary functionality, such as processing and transmission power. Otherwise it registers with the nearest cluster-head and become members of that cluster.

Due to the dynamic nature of mobile nodes in ad-hoc networks, their association and dissociation to and from clusters perturb the stability of the network and thus reconfiguration of cluster-head is unavoidable. This is an important issue since frequent cluster-head changes adversely affect the performance of other protocols such as security, routing and resource allocation that rely on it.

Choosing cluster-head optimally is a very hard task, to preserve resource of the network nodes. So any clustering scheme should preserve its structure as long as possible when nodes are moving around the network, to avoid overhead due to messages exchanged to deal with the topology changing.

1-1 Advantages

In this section we are going to cite a list of no exhaustive advantages of clustering, which have made it target of lot of recent research in ad hoc networks.

- Clustering facilitates the reuse of resource, which can improve the system capacity. In the way that information is stored once on the cluster-head [50, 51].
- Clustering can be used to reduce the amount of information that is used to store the network state [50, 51]. The cluster head will collect the state of nodes in its cluster and built an overview of its cluster state, which reflect the network state. Distant nodes outside of the cluster usually do not need to know details of specific events occurring inside the cluster [57, 58].
- Clustering may optimally manage the network topology, by dividing this task among specified nodes (cluster heads)[42].
- Cluster architecture may simplify the management of ad hoc network, for example clustering for routing reduces significantly the overhead costs imposed by routing [53]. Every node in the network is identified by its identity and the identity of the cluster to which it belongs. This architecture may reduce significantly the number of entries in the routing tables; an example of this is ZRP (*zone routing protocol*)[14], CBRP (*cluster based protocol*)[15]. Other adaptations using clustering have been recently given for existed routing algorithms as DSR [54, 55].
- With clustering an ad hoc network gets a structured topology as a cellular system in which the cluster head plays the role of the base station and can be used as a controller for security[31, 32], routing [53] or power controller [56]. The cluster head can choose specified node to help him in the management of these tasks; these nodes are called Gate Ways.

1-2 Criteria on clustering algorithm

Regarding the advantages of clustering for ad hoc networks, lot of algorithms have been proposed in literature, however the majority of them don't treat all aspects that must be taken into consideration by any clustering algorithm. In this section we'll cite some of these aspects:

- The algorithm must minimize the number of clusters by considering group mobility pattern [60].
- The algorithm must be distributed and executed asynchronously [43].
- The algorithm must incur minimal clustering overhead, be it cluster formation or maintenance overhead [49, 50, 51].
- Network-wide flooding must be avoided [57, 58].
- Optimal clustering may not be achieved, but the algorithm must be able to form the best clusters, according to their stability [43, 60].
- Clustering algorithm should be able to maintain its cluster structure as stable as possible while the topology changes [49, 50, 51].
- Clustering algorithm should achieve any cluster management operation with the minimal computational complexity.
- Clustering must use as possible all existed information of other underlying protocols (routing, security…).
- The cluster-head must be the node with the greatest capabilities, and must be in the center of its cluster (having the maximum neighbours) [43].
- A certain threshold of security must be considered when electing cluster-head[51].

2- State of the art

In this section we are going to present an outgoing of recent research and most known clustering algorithms for ad hoc network, we'll also give a brief analysis of each algorithm allowing us to identify the limitations of each one:

2-1 Highest-Degree Algorithm

The Highest-Degree Algorithm, also known as *connectivity-based clustering algorithm*, was originally proposed by Gerla and Parekh in [44], it's based on the degree of nodes assumed to be the number of neighbours of a given node

The authors suppose that every node has a unique Id, which is broadcasted within its neighbourhood. Each node x is considered as a neighbour of y if it is in the transmission range of y. after a given period, every node counts the number of received Ids (neighbours), then the node with the maximum neighbours (degree) is chosen to be cluster-head.

The neighbours of a cluster-head become members of that cluster and can no longer participate in the election process. Since no cluster-heads are directly linked, only one cluster-head is allowed per cluster. Any two nodes in a cluster are at most two-hops away. Basically, each node either becomes a cluster head or remains an ordinary node (neighbour of a cluster head).

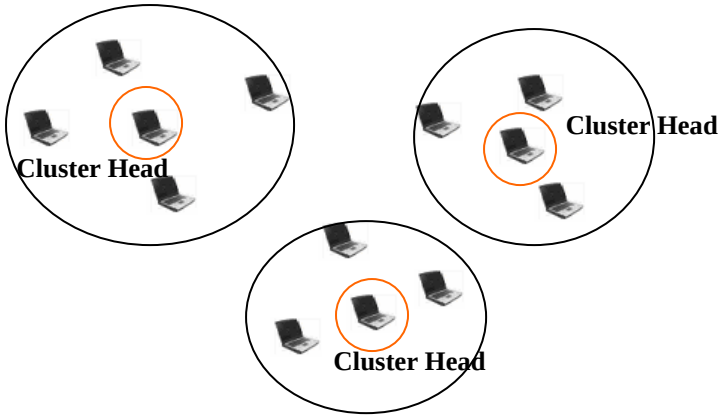


Figure IV.1 One Hop Cluster

Analysis

According to the criteria cited in section 1.2 we can conclude the following limitation of this algorithm:

- Typically, each elected cluster-head has a limited amount of resources (memory, CPU, battery…), which are not sufficient to serve a great number of neighbours, since this algorithm has not any restriction on the upper bound of the number of nodes in a cluster, which must be defined according to remainder of resource (memory, battery…). Because in some situation a node may be the highest degree node but its resources aren't sufficient to serve this great number of nodes.
- Another problem occurs when a cluster head loses one of its neighbours (cluster member), so it may not be re-elected, since it has not the sufficient number of neighbours. Resulting on a new election in that neighbourhood to elect another cluster-head. However the election procedure must be avoided with insignificant changing on the topology.
- It doesn't give any restriction on the lower bound of the number of nodes, which result on some cluster with a small number of nodes. Resulting on great number of clusters, which may increases the complexity of managing them when elaborating the virtual backbone between cluster-heads. To overcome this we can merge these clusters in one cluster with reasonable number of nodes.
- This algorithm creates one hop clusters, which is too small for large ad hoc networks, resulting on great number of clusters, which isn't desired for some underlying protocols like security, where cluster-head services aren't needed permanently.
- It doesn't treat the aspect of security.
- It uses flooding to broadcast all information to manage the topology maintenance which isn't desired and must be avoided.

2-2 Lowest-ID algorithm

The Lowest-ID Algorithm, also known as *identifier-based clustering algorithm*, was originally proposed by Baker and Ephremides [59]. This algorithm assigns a unique Id to each node and chooses the node with the minimum Id as cluster-head. Whenever a node with a

lowest Id is detected in the cluster, cluster-head must delegate it responsibility to this node to be cluster head.

A node is called a *gateway* if it is in the transmission range of more than one cluster-head. Gateway nodes are generally used for inter-cluster services, allowing the rely between clusters.

Analysis

Considering the election exigency we can say that:

- For this heuristic, the system performance is better compared to the Highest-Degree algorithm [43].
- Since the environment under consideration is mobile, with limited resource, it is unlikely that node degrees remain stable. Otherwise it must be counted according to the remaining of resource on each node, to elect the one with the greatest capabilities as cluster-head.
- In this algorithm Ids are arbitrarily assigned without considering the qualifications of a node possibly being elected as a cluster-head.
- This algorithm has also the same problem as the highest degree algorithm, since it creates one hop clusters which is too small for large ad hoc network.
- It doesn't treat the aspect of the upper bound of the number of cluster members in the cluster which must be limited by the remainder of resource on the cluster head.
- It doesn't limit the lower number of cluster members to minimize the complexity due to managing great number of clusters.
- It doesn't treat the aspect of security.
- It uses flooding, to manage the clustering structure.

2-3 Mobility-based d-Hop Clustering Algorithm

The authors in [60] propose a clustering scheme based on the real distance between nodes. They propose to calculate an estimate value of the distance between nodes by measuring the received signal strength taken from periodic beaconing or hello messages used in some routing protocols. According to this estimated value we can estimate the stability of every node. So we can decide to elect it cluster-head or not.

In order to compute mobility the author proposes a method to measure the real distance between nodes based on the signal strength using the following equation:

$$P_r = P_t \times G_r \times G_t \times \frac{\lambda^2}{(4 \times \pi \times d)^2}$$

Where

- *Pr* is the received power,
- *Pt* is the transmitted power,
- *Gt, Gr* are respectively the antenna gain of the transmitter and receiver,
- λ is the wavelength.
- *d* is the distance.

They also suppose that both the transmitter and the receiver have the same transmission power and having the same antenna gain. Then they use these suppositions to conclude a constant K to calculate an estimated distance between two nodes A and B using the following formula:

$$E[D_{AB}] = \frac{k}{\sqrt{P_r}} \text{ where k is constant}$$

Relative mobility between nodes A and B, indicates whether they are moving away from each other, moving closer to each other or maintain the same distance from each other. To calculate relative mobility, we compute the difference of the distance at time, *t* and the distance at time, *t* - 1. Relative mobility at node A with respect to node B at *t* is calculated as follows:

$$E[D_{AB}] = E[D^{t}{}_{AB}] - E[D^{t-1}{}_{AB}]$$

The variation of $E[D_{AB}]$ over a time period, T, VD_{AB}, is defined as the changes of estimated distances between node A and B over a predefine time period. Let's consider node A as a measuring node. Node A has a series of estimated distance values from node B measured at certain time interval for *n* times, $E[D_{AB}]=\{E[D_{AB}]t, t = 0, 1, 2, \dots, n\}$. Therefore we calculate VD_{AB} as the standard deviation of distance variations as follows:

$$\mathrm{VD}_{\mathrm{AB}} = \sigma(\left|E[D_{AB}]_1 - E[D_{AB}]_0\right|, \left|E[D_{AB}]_2 - E[D_{AB}]_0\right|, \ldots, \left|E[D_{AB}]_n - E[D_{AB}]_0\right|)$$

Local stability at node A, *St*A, represents the degree of stability at node A with respect to all its neighbours. Local stability is the standard deviation of relative mobility values of all neighbours. Therefore it is calculated as follows:

$$\mathrm{St}_{\mathrm{A}} = \sigma(VD_{AB1}, VD_{AB2}, \ldots, VD_{ABn})$$

After have computing the stability of each node desiring to be cluster-head, the most stable node is chosen as cluster.

Another proposition concerning the size of cluster is treated in this algorithm, since all existing clustering algorithms form two or one hop clusters which are not useful for large networks, the authors proposes to form *D* hop clusters, to serve efficiently the underlying protocol. Since the parameter *D* can be changed any underlying protocol can change it to serve its needs, according to the size of the network, or the amount of services needed from the cluster-head.

Analysis

According to the criteria defined in section 1.2 we can observe the following inconvenient:

- This algorithm creates D hop clusters which can be changed according to the underlying protocol. However the size remains the same and doesn't change whenever the number of cluster members increases or decreases.
- It treats neither the upper nor the lower bound of the number of cluster members. Resulting on clusters with unreasonable number of nodes which causes cluster-head to be busy quickly (this is proven by simulation).
- This algorithm include in the election the aspect of nodes stability in order to elect the most stable node, however the method used to compute this stability include unrealistic supposition like the supposition on the antenna gain and transmission power which is assumed to be the same. This assumption isn't always true in ad hoc networks characterized by the heterogeneity of its nodes. He also uses complex computing to compute stability which isn't desired in ad hoc network where mobile nodes are with limited power computing.

- It doesn't treat all aspects of mobile nodes characteristics like battery power, and transmission range.
- It doesn't treat the aspect of security.
- It uses flooding for broadcasting packets to manage cluster structure.

2-4 Weight base Clustering Algorithm (WCA)

This algorithm was proposed by SAJAL and TURGUT in [43]; they begin from the supposition that the algorithms cited above don't treat all the characteristics of ad hoc networks that influence on the cluster-head election. Since every one of them is intended to work for some specific situation of ad hoc networks. For this purpose they have proposed a weight based algorithm, which means that the cluster head is elected according to its weight, which is calculated by combining a set of system parameters (battery, mobility, and distance between nodes…). Depending on the underlying protocol routing or security, any or all parameters can be used. The parameters are combined with a certain weighting factors given depending on the importance of this parameter for the underlying protocol.

The author proposes also to use an upper bound which limits the number of nodes served simultaneously by cluster head.

Cluster head election algorithm: The network formed by nodes and links can be represented by an undirected graph $G = (V, E)$, where V represents the set of nodes vi and E represents the set of links ei .

The cluster head election consists of eight steps:

Step 1: Find the neighbours of each node v (i.e., nodes within its transmission range) which defines its *degree*, d_v, as

$$d_\upsilon = \left| \sum_{\gamma \in V} dist(\mathrm{v}, \gamma) < \mathrm{tx}_{\mathrm{range}} \right|$$

Where $\mathrm{tx}_{\mathrm{range}}$ is the transmission range of v.

Step 2: compute the degree deference for every node v as:

$$\Delta_\nu = \left| d_\nu - \delta \right|$$

Where δ is the maximum number of nodes that this cluster-head can support.

Step 3: For every node, compute the *sum of the* distances, D_v, with all its neighbours, as

$$D_\upsilon = \sum_{\gamma \in N(v)} dist(\mathrm{v}, \gamma)$$

Step 4: Compute the running average of the speed for every node till current time T. This gives a measure of mobility and is denoted as M_v, as

$$M_\nu = \frac{1}{T} \sum_{t=1}^{T} \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$$

Where (X_t, Y_t) and (X_{t-1}, Y_{t-1}) are the coordinates of the node v at time t and (t-1), respectively.

Step 5: Compute the cumulative time, P_v, during which a node *v* acts as a cluster-head. P_v implies how much battery power has been consumed which is assumed more for a cluster-head than an ordinary node.

Step 6: **.** Calculate the combined weight W_v for each node *v* as

$$W_v = W_1\Delta_v + W_2 D_v + W_3 M_v + W_4 P_v$$

Where W_1, W_2, W_3 and W_4 are the weighing factors for the corresponding system parameters.

Step 7: that node with the smallest W_v as the cluster-head. All the neighbours of the chosen cluster-head are no longer allowed to participate in the election procedure.

Step 8: Repeat steps 2–7 for the remaining nodes not yet selected as a cluster-head or assigned to a cluster.

Analysis

This algorithm takes into account a lot of parameters, thus it may be a good alternative of our security protocol. Since the author of this protocol has proposed that the set of parameters may change from an application to another, according to the needs of the underlying protocol. However it presents some inconvenient:

- It creates one hop cluster which is too small for large ad hoc network.
- Like other algorithms it uses flooding, and it doesn't include the aspect of security in the election.
- The authors of this algorithm try to include the aspect of mobility in the election procedure in order to elect the most stable node, but their method to achieve this computing is based on the real coordinates of nodes which are too difficult to compute without the existence of GPS (Global Position System) adapters on each node in the network. GPS adapter give the exact coordinates of nodes and it is largely used nowadays, however these adapters are not democratised and used by all mobile node including existed phone and other small PDA, it also takes long time to compute the exact position. We can also observe a degradation of service if nodes move quickly.
- Another problem is the complex computing performed to compute mobility which isn't adequate for mobile node characterized by limited resources.

2-5 double manager k-hop clustering algorithm in mobile ad hoc networks

The authors in [61] propose a new routing algorithm based on clustering. Our interest in this routing algorithm is the underlying clustering algorithm. Therefore, the authors propose a clustering algorithm based on link evaluation. This algorithm is also a weight based clustering algorithm, since it affects to each link between a given node and its neighbours a weight according to the signal strength received from each neighbour. In this way each node gets its global weight by summing the partial-weight of each link, we note that the great weight is given to the link with the greatest signal strength, which is assumed to be the nearest node. Then the node with the greatest weight is elected as cluster-head.

The above explication describe one hop cluster formation, however when creating more than one hop clusters, the weight computing is changed by using a weight factor to each set of neighbours. The greatest weight factor is given to one hop neighbours, and the lowest is given to farther nodes. In the same way as described above each node compute the link weight of its one hop neighbours, and then he gets information from its neighbours about two, three … hops links. After getting all information until k hops, where k is the radius of the cluster, each node combines link-weights with the corresponding weight factors and get its global weight, according to this global weight a node is elected cluster-head or not.

According to these global weights each node is considered as gateway, member or sub-head to be used in routing. A sub head node is a node helping the cluster in intra-cluster management, which justifies the name of this clustering algorithm, since there are two managers of clusters (cluster-heads and sub-heads).

Analysis

- The authors present a weight based clustering algorithm, but they don't include any system parameter as CPU power, Memory or battery to elect cluster-head.
- They assume that nodes have the same transmission power to permit the link evaluation, which isn't true in ad hoc networks.
- The aspect of nodes mobility isn't included to elect as cluster-head the most stable node.
- They don't include the aspect of security on the election criteria.
- The authors don't treat the aspect of upper bound of the maximum node in clusters.
- This algorithm is based on link evaluation according to the received signal strength, allowing the algorithm implicitly to elected as cluster-head the node with the maximum neighbours because one hop link are given the greatest weight and weight factor.
- The network overhead is low compared to other algorithms.
- There election procedure is simple and very possible to be used in ad hoc networks.

2-6 Conclusion

As we can observe lot of propositions are given in literature, giving solution to only some specific problems of ad-hoc network, however none one treats the entire characteristics of ad hoc network (mobility, transmission range, size of the network, capabilities of the nodes…).

For example the highest-degree, lowest-ID algorithms creates one hop cluster, which is too small for large network resulting on lot of clusters, which complicate the virtual backbone management. They are also sensible to small changing in the topology.

The WCA and Mobility based algorithms try to include the mobility (stability) of nodes as a factor in the election procedure, in order to elect as cluster-head the most stable node. But their methods to compute stability are based on some assumptions which are not always valid in all ad hoc networks.

The mobility based algorithm tries to solve the problem of the small size of clusters encountered in other algorithm by creating D hops clusters. However it doesn't treat the aspect of the maximum number of nodes in a cluster, because cluster-heads can't serve infinite number of nodes.

Another observation that must be made is the aspect of security, which isn't included in any one of the above algorithms. As it's presented in [16, 17, 18] the problem of security is a pertinent problem, so it must be taken into consideration in all schemes defined for ad hoc networks.

3- Secured Clustering Algorithm SCA

In this section we'll give an overview of our proposed clustering algorithm called Secured Clustering Algorithm (SCA). In which we include the aspect of security by using trust value and certificate to overcome the limit of security in existed clustering algorithm making in this way our algorithm useful for security purpose as PKI based on clustering [31, 32]. SCA is a weight based clustering algorithm uses weight computed from a set of parameters, to elect the node with the maximum weight, including a new method to compute stability.

We'll define a set of cluster and nodes management operations intended to maintain the stability of the clusters.

We also propose to use efficient flooding to broadcast any information over the network.

3-1 Basis for our algorithm

Every clustering algorithm, tries to elect as cluster head the node with the maximum capabilities, in order to serve its CMs as long as possible, it also tries to maintain the stability of clusters. In this section we give a brief definition of all aspects included in SCA to guaranty these two criteria:

3-1-1 Cluster management criteria

- *Cluster Size:* some of the algorithms cited above, define the size of the cluster depending on transmission range of each node, by constructing one hop clusters, which is too small for large network. To overcome this SCA creates D hops clusters, where D is modified according to the number of the cluster members and nodes capabilities by executing cluster size extension and reduction operations.
- *Cluster division:* each *CH* can ideally support only *Max* nodes to ensure efficient services. The value of *Max* is defined according to *CH* capabilities (Memory, battery…). If the *CH* tries to serve more nodes than it is capable of, the system efficiency suffers in the sense that the nodes will incur more delay because they have to wait longer for their turn to get their share of the resource. So the definition of the threshold *max* as an upper bound of the maximum of node in a given cluster may solve this problem, so whenever the number of nodes exceeds the threshold of *Max* the CH must launch the cluster division mechanism to divide the cluster into two small clusters, so that every CH serves efficiently its cluster members. The value of *Max* is defined according to the capabilities of the cluster-head.
- *Cluster size reduction:* This operation is executed after the division of the cluster, intended to reduce the radius of the cluster from D to D-1, which means that beacons don't reach the boundaries of the cluster, resulting on the roaming of boundaries nodes to other clusters including new created cluster which may reduce the number of nodes in that cluster.
- *Merging Cluster:* since the point of failure of any cluster based security mechanism is the CH, where are kept all important information, so we must minimise as possible the number of CHs in the network. In order to accomplish this and to minimize the complexity due to the management of lot of clusters we prefer to merge clusters with few nodes into other cluster, in the sense that all CMs of the old cluster become members of adjacent clusters. To launch the cluster merging process we define a predefined value *Min* as a lower bound of the number of nodes that may exist in a given cluster before merging it with an adjacent cluster. The mechanism of cluster merging may minimize the complexity due to the management of great number of clusters when elaborating backbone network.
- *Cluster size extension:* This operation is executed whenever the merging procedure isn't successfully executed, thus the CH proceed to the extension of the radius of the cluster from D to D+1. Therefore beacons are broadcast within largest area allowing new nodes to join the network which may increase the number of nodes in that cluster.
- *Roaming:* ad hoc networks are characterized by dynamic topology, nodes move around the network, to go far from their original CH. In this situation a roaming mechanism must be executed to detach it from the old CH and attach it to the nearest CH.
- *Trust value:* it defines how much a given node is trusted by the network members, it is used when electing CH to choose the node with the greatest value to be CH, in this way we can exclude misbehaving nodes from the election procedure. Trying with this factor to keep network in a secured state by electing the most trusted nodes as CHs.

3-1-2 Electing criteria

- The *battery power:* A CH consumes more battery power than an ordinary node since a *CH* has extra responsibilities to carry out for its CMs. This aspect must be taken in account when electing CH, to elect the node with the greatest amount of remaindering power in its battery, allowing it to serve long.
- *Stability:* this is an important factor in deciding the CH. It isn't desirable to elect a CH that moves very quickly, resulting on a lot of cluster changes, occurring when the CH goes far from the area where there are its CMs. So all CMs must be detached from this CH, to be attached to another CH. Since this operation is very costly due to the amount of data exchanged, it must be minimize as possible, by choosing the appropriate CHs, which must be the most stable node in the network.
- *Trust value:* this value defines how much any node is trusted by the rest of the network nodes, according to the amount of this value a node is considered as trust or not, so it may be or not elected as CH.
- *Degree*: is the number of neighbours of a given node, within a given area. This parameter is defined to choose as CH the node with the maximum neighbours to serve more number of nodes.
- *The Max Value*: This criterion is used to elect node which can handle the maximum of node, since every node can serve efficiently a predefined number of node according to its resources, therefore it's desired to elect as CH the node which can serve the maximum of nodes.
- *Weight Factors*: each of the preceding parameters is called partial weight. Each one is affected a weight factor defining its degree of importance for the underlying protocol or the network, giving in this way more flexibility and large scale of use for our algorithm. Since only a subset of the parameters cited above can be used according to the exigency of the network and the underlying protocol. For example trust value may take the great value if the underlying protocol is a key management protocol. Factors are given values between 0 and 1, so the sum of factors is 1.

$$\sum_{i=1}^{n} F_i = 1$$

n is the number of factors

- *Global Weight*: using the parameters cited above every node in the network computes its global weight by combining each partial weight with the corresponding weight factor. Depending on the value of this weight a given node can be elected CH or not.

We denote W_T, W_D, W_B ,W_M,W_S the partial weights and F_T, F_D, F_B ,F_M,F_S are the weight factors corresponding respectively to (Degree, Battery, Max Value, Stability). The global weight is computed as follow

$$W_G = F_T \times W_T + F_D \times W_D + F_B \times W_B + F_M \times W_M + F_S \times (-W_S).$$

3-1-3 Node status

In a clustered network any node has one of the following statuses:

- *Non determinist status (N): this* status is affected to any node which isn't attached yet to any cluster, thus it must execute the Init procedure to attach him to one of the existing clusters.
- *Cluster-head (CH):* this status means that the corresponding node is a cluster-head.

- *Ordinary-Node (ON):* any node that isn't a CH neither a gateway is an ordinary node also called cluster-member (CM); it's attached to only one cluster, this cluster can be changed when the node roams from one cluster to another.
- *Gateway (GW):* we call a node that belongs to two or more zones of different clusters at the same time as gateway. A gateway can hear beacons sent from the two clusters, but it's attached to only one of them. The definition of the gateway is very useful for some underlying protocols like routing, since it can relay between the clusters to which it belongs. A mechanism to declare and choose gateways may be defined to choose ones with best profile.
- *CH-Ready:* this state is affected to any node desiring to be CH during the election procedure, and it is broadcasted to all neighbours to inform them that the corresponding node is ready to be CH.
- *GW-Ready*: this state is affected to each node in the transmission range of two clusters, therefore this node send to its CH *GW-Ready* beacon informing him that this node is ready to be gateway, depending on the cluster needs the CH decides to elect it as gateway or not.

3-2 Computing trust value

As we have mentioned above a trust value defines how much this node is trusted by the rest of the network nodes. So trusty nodes may have the greatest value, therefore we can use them to perform some special task like CH. An estimate value is calculated for a given node by its neighbours, according to observations made on the behaviour of the node. In this way we may exclude malicious nodes which must have the lowest trust value in the network.

In order to compute the trust value we define a voting mechanism in which a node *i* needing the trust value defused a trust value request within the radius of D hops, including its identity assumed to be its certificate *{Trust_Value_Request(ID(i))}*. According to the behaviour of the node *i* observed by *j* the node *j* returns a reply including the trust value $Tr_{j,\,i}$ *{Trust_Value_Reply(var)},* estimated by *j* to *i.*

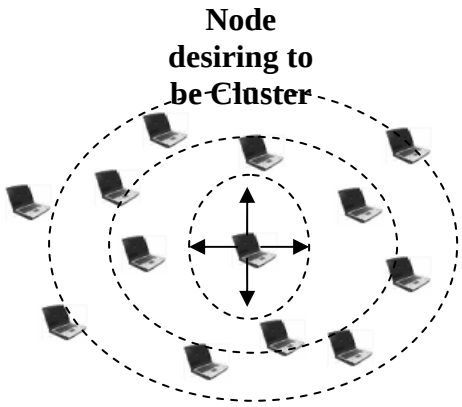


Figure IV.2 Request for trust value

When receiving values from all neighbours or whenever the delay of trust value computing is expired, node *i* can calculates the estimate trust value T, as the average of all received values from its neighbours.

$$T = \frac{\sum_{i=1}^{N} T_i}{N}$$

3-3 Computing The Stability (mobility)

The stability is an important parameter when electing the cluster-head. In order to elect the most stable node as cluster-head, because electing no stable nodes may affect the stability of the clustering architecture.

The stability is defined as the property of a node to be as long as possible within the same neighbourhood, in the way that this node move with its neighbourhood in the same direction or it stays in the same zone with the majority of its neighbourhoods.

The stability of a given node is defined during a time period T in which a node was stable. In order to compute stability we define the following terms:

1- *The distance*: the distance between two nodes A,B ($D_{A,B}$), is defined as the number of hops between them, which can be obtained from the packets sent from one to the other, or hello message used in routing protocols. The possibility of obtaining the number of hops between two nodes is evident and simple with all existed routing protocols. We have proposed to compute the distance using the number of hops to overcome the limitations of the methods defined by other algorithms cited above. In the sense that this method is simple to compute and it doesn't include any complex operations, and the distance is obtained from hello and control messages sent by other protocols, by observing their TTL (Time To Live) field in IP datagramme.

2- *The mean distance*: it is defined as the average of distances between node A and all its neighbours, within D hops radius.

$$\mathrm{MD}_{\mathrm{A}} = \frac{1}{N}\sum_{n=1}^{N} D_{A,n}$$

N is the degree of A

MD takes values between 1 and D and it defines the radius where there exists the great density of nodes. For example when MD=2 this means that the majority of neighbours are within 2 hops.

3- *Stability*: is defined as the difference between two measures of MD at t and (t-1), it is large when the node goes far from its neighbours or whenever its neighbours are going in other direction than the one taken by the considered node. The value is compared with D and a node is considered as most stable if it has the less value of ST.

$$ST_A = MD_t - MD_{t-1}$$

3-4 Beaconing

Beacons are messages sent by CHs within a given area, in our case within D hops. These beacons are broadcasted periodically and are received by nodes which are at D hops from the CH. Beacons are used to:

- Inform nodes about the existence of a cluster-head in the area.
- Send commands to cluster members to collaborate in execution of cluster management operations.
- During the election procedure.
- Computing trust value and stability.
- Other actions needed by the underlying protocol like routing if it's a cluster based routing protocol.

Beacons must be received by the CMs and any other nodes moving in the zone of this CH. It must contain all information about the CH.

3-4-1 Mechanisms of beaconing

To broadcast beacons we can use one of the following mechanisms:

a. Multicast: this mechanism is used to send packet p to a group of nodes in the network, and it's known to be efficient. One of known techniques to achieve multicast is by using unicast with every node in the group. This solution isn't desired in our scheme because beacons must be received by CMs and other nodes moving in the same zone. However in our case the CH has the list of its CMs but he can't get the list of visiting nodes, which make this solution undesired for beaconing.

b. Flooding: In flooding, a node transmits a message to all of its neighbours. The neighbours in turn relay to their neighbours and so on until the message has been propagated to the entire network [57]. This mechanism is used by the majority of routing protocols to discover paths. Flooding can be used for our scheme because it guarantees that beacons are received by all nodes within the desired zone.

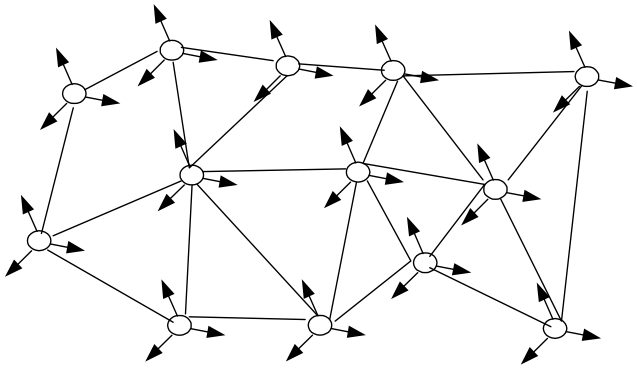


Figure IV.3 The flooding in ad hoc network

As we can easily see, the performance of flooding is closely related to the average number of neighbours. As the neighbour degree gets higher, flooding suffers from the increase of

(1) Redundant and superfluous packets: in flooding every received packet is sent to all neighbours, however the majority of neighbours have already received this packet, which increases the number of redundant packets in the network.

(2) Probability of collision: since the packets flooded are sent by all nodes in the same time this augment the risk of collision which degrades the performance of the network.

(3) Congestion of wireless medium: Packets flooded lock medium for a certain period. So any other control messages aren't sent which embarrass other protocols.

(4) Security risks: the majority of the algorithms used to encrypt traffic over the network are symmetric algorithms. These algorithms can be cracked when gathering the necessary number of packets. Therefore flooded packets can be used to construct data-base to store interesting packet used later to crack the algorithm and determine the key. For example the RC4 algorithm (RC4 is largely used in wireless network) can be broken when gathering some millions of packets. This can be achieved quickly when taking advantages of flooding, because the number of packet is very high in the flooding period. The WEP protocol used in 802.11 can be cracked after 6 hours if the traffic sent over the network is

very high, so it is impossible to use a similar protocol in ad hoc network if we keep the mechanism of flooding.

Flooding is a very important mechanism in Ad-hoc network. However the main problem with flooding is the number of packet sent in the period of flooding which is very high. In order to keep it efficiency we must minimize the number of packet during flooding. This is accomplished by using efficient flooding.

c. Efficient Flooding: as we have said in the above section, the mechanism of flooding is not desired to broadcast beacons messages. An evident solution to this problem is to minimize the flood packet circling in the network during the period of flooding; this can be done efficiently by minimizing the number of nodes included in flooding. Given that the number of nodes relay flooded packet are kept within a little threshold for a given size of neighbourhood, this may minimize the effect redundant packets. In [57] and [58] the authors specify different heuristics where a node decides whether to relay or not a flooded packet based on one of the following heuristics:

i. Rebroadcast with a given probability this solution is based on probability in the sense that the node compute the probability to participate in flooding or not based on the number of neighbours, and the number of redundant packet received in the period of flooding.

ii. Rebroadcast if the number of received duplicate packets is less than a threshold: this solution is based on the idea that the number of duplicate packet is relative to the number of neighbours, therefore if the number of duplicate packet is high this means that the number of included nodes in the flooding is high so it must exclude itself from the operation, however if the received packet is little this means that there is not lot of nodes participating in flooding, so it must relays the received packets. In this manner the number of nodes is kept around an acceptable threshold allowing an efficient flooding.

iii. Location-based scheme where the relative distance between hosts determines the rebroadcast decision: this solution is based on the pre-knowing of the exact distance of every node in the network. So when receiving a flooded packet a node decides to relay it if the sender node is far from it, on other hand the node doesn't relay it.

iv. Cluster-based scheme: this is a cluster based solution where only CHs and gateway participate in flooding, since the number of CHs and gateways is limited within the network, this limits the effect flooding on the network. This solution may be very useful if the cluster architecture is already used for other purposes like in our solution. However if the clusters are created and maintained only for flooding this may incur additional costs.

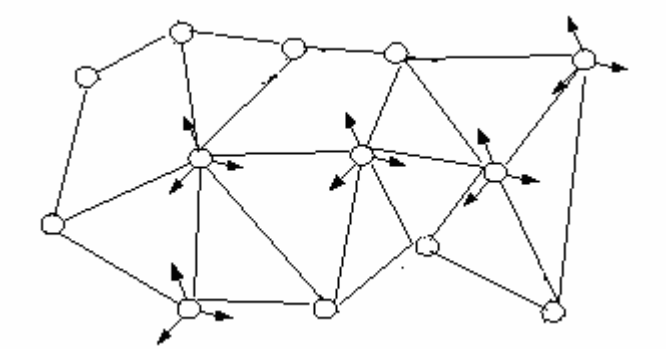


Figure IV.4 The effect of efficient flooding

Experiment has proven that the performance of flooding is better when using any one of the above propositions, so we don't precise any mechanism to improve flooding, however it's suitable to use cluster based ones, since we are using clustering scheme we can profit from this and use it for flooding.

3-4-2 Structure of beacons

Like we have said beacons are messages broadcasted over the network allowing cluster maintenance. Nodes listen permanently to these beacons, in order to define their status and execute the corresponding tasks. The known use of beacons is to inform CMs by the existence of a CH in a given zone, thus we include in the beacons, the identity of the CH and we don't need any additional information. However in our algorithm we have added some additional mechanisms as cluster merging or division, needing the collaboration of all the CMs. To achieve this we have add new field containing the code of the action demanded by the CH to be executed in collaboration with the CMs. This two criteria give to Beacons the structure in Figure III.5.

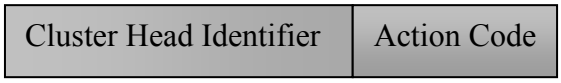


Figure IV.5 : Beacon structure

a- Identity field

This field contains the identity of the CH, in literature there are several mechanism used to identify the CH:

Using IP or MAC address: if we use one of the two addresses we don't need to include this identity in the beacon message, because these two addresses can be obtained from the received packets which include the beacons. Using IP or MAC address isn't desired because it can be spoofed by malicious nodes.

Certificate: Whenever the underlying security mechanism is a PKI infrastructure it's suitable to use the certificate of the CH to identify it. In the previous chapters we've presented several PKI infrastructure including our contribution which is based on clustering.

b- Action code field

This field is used essentially for maintenance, because the CH must collaborate with its CMs to maintain the network in a true status. In our algorithm we have justified the use of some additional actions like cluster merging or devising … thus we must define for each action a predetermined code publicly known in the network allowing CMs to collaborate in order to execute the needed action. We must also give this field an extensible area used by the underlying protocol to execute its appropriate actions for example in a security protocol we can define a code for key update if the encrypting key is updated periodically.

3-5 Cluster-Head election procedure

This operation is invoked whenever a neighbourhood is without a CH. The invocation of the election procedure doesn't mean that all CHs are replaced. It can be launched at any time along the lifetime of the network.

Like we have said SCA constructs D hops clusters, so any broadcasted information is diffused within D hops. To accomplish this we use the efficient flooding described above.

We assume that a set of node desiring to create or to maintain a clustered (hierarchical) structure so they must collaborate to execute the following steps:

3-5-1 Discovery stage

The purpose of this step is to get information about the neighbourhood where the election procedure is invoked. Thus nodes desiring to be CH send cluster-head_ready beacons within the radius of D hops. Each node when receiving these beacons estimates a trust value and sends it back to the asking node. After the discovery period T_d, nodes have initiated this operation can conclude from the received responses the following information:

-Degree (W_D): This is the number of received response.

-Stability (W_S): concluded using the method described in section 3.3.

-Trust value (W_T): computed using method described in section 3.2.

3-5-2 Computing weight

After the achievement of the discovery stage each node adds to the con concluded weights the state of its battery (W_B) and the maximum value (W_M) and combines them with the corresponding weight factors F_D, F_S F_T, F_B ,F_M to conclude the global weight.

$$W_G= F_T\times W_T +F_D\times W_D+F_B\times W_B+F_M\times W_M+F_S\times(-W_S).$$

This weight is broadcasted within the same neighbourhood, in order to elect CH. Nodes receiving different weights choose as CH the node with maximum weight, and attach themselves to this CH.

3-5-3 Elaboration of the backbone

Whenever the previous step is successfully achieved, each elected CH need to discover each other to elaborate a virtual backbone to ensure inter-cluster services. Thus every new elected CH broadcast a discovery request over the entire network; CHs receiving this request register the certificate of the new CH and send him a response containing their certificate.

3-5-4 The algorithm

The following algorithm shows the steps executed by each node during the election procedure. First the node changes its status to CH-Ready status (Line 2), and then it broadcast this status to launch the election procedure (Line 3). After performing the needed initialisation (Line 4-6), the node loops for a given period T_E (Line 7-12) allowing him to compute the sum of received trust value (Line 10), and its degree (Line 11). After the achievement of the loop the node adds to its trust value and the degree the state of its battery, and the stability (Line 13-18) to compute the global weight W_G (Line 19). The global weight is then broadcasted (Line 22) in the same neighbourhood to allow the election of the CH.

Procedure Election( )

```
1.  Begin
2.  this.Status= CH-Ready;
3.  Broadcast(CH-Ready);
4.  Wait();
5.  Sum_Trust=0;
6.  Degree=0;
7.  While T do
8.    begin
9.       Receive (Response);
10.      Sum_Trust= Sum_Trust+Response.Get_Trust_Value;
11.      Degree= Degree+1;
12.   end;
13. Trust = Sum_Trust / Degree;
```

14. W_T =Trust;
15. W_D =Degree;
16. W_B =Get_Battery_State();
17. W_M =get_max_nodes(this);
18. W_S =Stability(this);
19. $W_G = F_T \times W_T + F_D \times W_D + F_B \times W_B + F_M \times W_M + F_S \times (-W_S)$.
20. this.Weight= W_G;
21. Broadcast (Identity(this), this.Weight);
22. End;

3-5-5 Algorithm executed by CM

Like it was mentioned above during the election procedure all nodes of a given neighbourhood collaborate to accomplish the execution of the election procedure. Thus they must execute the following procedure in response of certain requests, and to define their CH.

a- When receiving request for trust value

When receiving this request the node estimates a trust value for the requesting node, and sends it back to the asking node. We assume that there is a mechanism allowing any node to observe the behaviour of its neighbours.

Procedure Quest_For_Trust( )

1. Begin
2. Trust=x;
3. Send (Trust,v);
4. End;

***b*- When receiving messages containing weight**

Whenever this message is received the node compares the received weight W_r with the stored weight W_s (Line 7) and chooses as CH the one with the greatest weight (Line 9), this operation is executed during a given period T, to ensure efficiency. This operation is executed by each node in the neighbourhood in which the election procedure is launched, in the end of this procedure the node determines its CH and joins that cluster by executing the JOIN command.

Procedure CH_Definition( )

1. Begin
2. CH=null;
3. W=0;
4. While T do
5. Begin
6. Receive (Cert,W_r);
7. If $W_r > W_s$ then
8. begin
9. CH=Cert;
10. $W_s=W_r$;
11. End;
12. End;
13. JOIN(CH);
14. End;

3-5-6 Security feature

To protect the election procedure from malicious nodes willing to insert false information during the election procedure, we propose that all messages sent during the election procedure are read by all neighbours, so that they can identify these malicious nodes, and reject any information coming from malicious nodes.

Another aspect is the protection against strange malicious nodes, thus we propose to encrypt exchanged data. In this way only the authorized nodes are allowed to participate in the election procedure.

3-6 Cluster maintenance

We define the cluster maintenance operation as the operation executed to maintain the structure of clustered network stable as long as possible. To accomplish this, network nodes must collaborate to handle the following operations:

3-6-1 Initialisation of nodes

The Init procedure is executed by each node in a no determinist status. A node with this status is a node which isn't attached yet to any cluster, this may be caused by a link failure, roaming, or whenever the node comes for the first time the network.

First the node listens if there is any neighbouring CH (Line 3) the function Get_beacons() returns the list of all received beacons. If this is the case it chooses the nearest cluster and joins it (Line 11, 12). Otherwise it launches the election procedure to elect new CH in this neighbourhood (Line 6).

Procedure Init ()

```
1.Begin
2.  If status=N then
3.  CH-list=Get_beacons();
4.  If CH-list=null then
5.        begin
6.          Election ( );
7.          exit();
8.        end;
9.  else
10.       begin
11.         CH=CH-list.Get_Nearest ( );
12.         JOIN(CH);
13.       end;
14.  exit;
15. End;
```

3-6-2 Receiving Beacons

This procedure is executed by every node including gateway and CH and ordinary nodes. However it has different effect when executed by any kind of nodes. According to the corresponding information contained in these beacons the node decides its following operation.

First the node listens if there is any neighbouring CHs (Line 2). If it doesn't receive any beacon and its status isn't CH, this means that this node has lost the link with its CH so it must changes its status to no determinist status and launches the Init procedure (Line 5, 6). Otherwise the node verifies if it listens more than beacon, which means that this node can be

used as gate way, so it changes it status to Gatway-Ready and sends an alert to all neighbouring CH (Line 14,15).

Procedure Receive_Beacons( )

```
1.  Begin
2.   CH-list=Get_beacons();
3.   If CH-list=null and this.Status ≠ CH then
4.     begin
5.       this.Status=N;
6.       Init();
7.     end;
8.   Else
9.    begin
10.    if this.CH in CH-list then
11.     begin
12.       if CH-list.length>1 then
13.                  begin
14.                        this.Status=Gatway-Ready;
15.                         send(alert, CH-list);
16.                  end;
17.     end;
18.  end;
19. End;
```

3-6-5 Cluster division

Like it was already mentioned a cluster division mechanism is launched whenever the number of nodes in a given cluster increases to be more then a predefined threshold defined according to remainder of resource of the CH (Memory, battery…)., or when the CH becomes busy because it has served as CH for a long time until it becomes incapable to serve the actual number of CMs. So it launches the election procedure in its cluster area to elect a new CH allowing the creation of a new cluster.

Therefore the CH broadcasts Cluster_Division request to its CMs (Line 4), whenever this request is received each CM compute its weight and sends it back to the CH, which saves them (Line 7). Then the CH chooses as new CH the farthest node with the maximum weight and sends him a grant response (Line 10). Then the new CH begins sending beacons and creates its own cluster.

Procedure Cluster_Division( )

```
1.  Begin
2.  If  Member > MAX then
3.  Begin
4.    Broadcast (Cluster_Division);
5.     Wait;
6.     While T_D do
7.       list= Receive (Cert, W);
8.      enddo ;
9.     Cert= List.get-max-weight.
10.    Send(grant,Cert);
11. endif;
12. End;
```

3-6-6 Cluster size reduction

This operation is executed after the division of the cluster, intended to reduce the radius of the cluster from D to D-1, which means that beacons don't reach the boundaries of the cluster, resulting on the roaming of boundaries nodes to other clusters including new created cluster.

3-6-7 Cluster merging

In the algorithms defined in literature no lower bound is defined to limit the minimum number of nodes in a cluster, resulting on some clusters with two or one nodes which is not suitable for large networks. Therefore in our algorithm we propose to merge such clusters immediately with the nearest cluster if it exists by executing the merging procedure.

First the CH listens if there are any neighbouring CHs (Line 4), if this is the case it broadcast a merging request (Line 7). Then it wait until receiving all confirmation from its CMs or the expiration of the delay T_M (Line 10-14) to choose the nearest cluster and roams to that cluster (Line 16).

Procedure Cluster_Merging( )

```
1.  Begin
2.  If Member<min then
3.    Begin
4.     CH-list=Get_beacons();
5.     If CH-list=null then
6.      exit (0);
7.     Broadcast (Cluster_Merging);
8.     Wait();
9.     i=0;
10.    While T_M or I< =Member do
11.     begin
12.       Receive(Confirmation);
13.        i++;
14.     end;
15.  if Member=0 then
16.          Roam(CH-list.Get_Nearest);
17.  End;
18. End;
```

3-6-9 Cluster size extension

This operation is executed whenever the merging procedure isn't successfully executed, thus the CH proceed to the extension of the radius of the cluster from D to D+1. Therefore beacons are broadcast within largest area allowing new nodes to join the network which may increase the number of nodes in this cluster.

3-6-10 Other scenarios

a- Roaming

This operation is executed whenever the node goes far from its cluster, so it must be detached from the old CH (change the status to no determinist), and attached to the nearest CH by executing the Init procedure.

b- Link failure

In this situation we assume that the contact with the CH isn't active for any causes, so the node permutes to no determinist status and executes the Init procedure.

4- Experiment results

In the section we are going to present our experiment results obtained by simulation. We include in our simulation two different algorithms to compare them with our SCA.
Mobility-based d-Hop Clustering Algorithm: electing the less mobile node as CH, and the highest degree algorithm: electing as CH the node with the maximum of neighbours.
In our simulation experience we have based the comparison on the number of clusters, the number of election and the number of CMs in each cluster.

4-1 Simulation environment

To evaluate the new clustering algorithm and compare it to existing algorithms, the ClusterSim simulator was used as a tool. This simulator is developed locally in our laboratory. This simulator is developed specially to evaluate clustering algorithms, since it implements all primitives that can be used in any clustering algorithm as the remainder of battery, the mobility, the distance…etc. However the election procedure must be written for every algorithm according to specification of each algorithm. In our case we have implemented three algorithms including our proposed algorithm.

The simulator gives as results the average of 20 simulations, and returns the number of clusters, the number of election, and the number of roaming requests

4-2 The scope of simulation

The purposes of the simulation are:

- To prove the unfeasibility of one hop clusters.
- To justify our choice to limit the number of nodes in clusters.
- To compare our algorithm with existed algorithms.
- Test the performance of SCA in dense networks (high number of nodes in a limited area).

To perform tests we've implemented three algorithms (Highest Degree Clustering Algorithm (HDCA), Mobility Based Clustering Algorithm (MBCA) and Secured Clustering Algorithm (SCA)). The scenarios were generated using parameters listed in table 1.

We vary the number of nodes from 25 to 300 nodes. The area is also varied from 100*100 m to 1000*1000m to allow the test for different configurations.

For SCA which include another aspect which is the upper bound of supported nodes, a random value between 20 and 30 nodes is affected for each node.

When launching simulation each node is affected a random position and a random speed less than 20 m/s. The simulation is executed during 300 s, during this time every node move randomly within the chosen area and change the direction whenever it reach to the boundaries of the area.

Parameters	Values
Network size	100*100m -1000*1000 m
Number of Nodes	25-300
Max speed	20 m/s
Pause Time	0 s
Transmission range	20 and 100m
Max of nodes in a cluster (used only for SCA)	Randomly chosen between 10 and 30
Simulation time	300s

Table IV.1 Simulation parameters

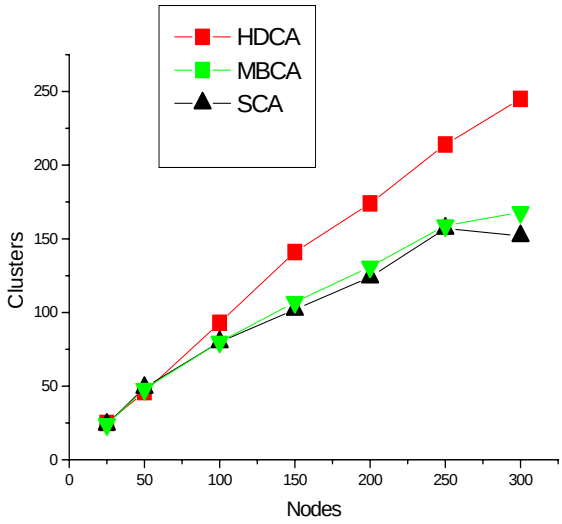
We note that for all the algorithms the number of clusters is relatively high when the transmission range is small or when the area is big. Because nodes are out of range of each other, therefore they form one or two node clusters. In this situation we observe that the results given by all algorithms are relatively adjacent, however the difference is clear when the number of nodes get high or whenever the area is too big or small.

4-3 proving the unfeasibility of one hope clusters

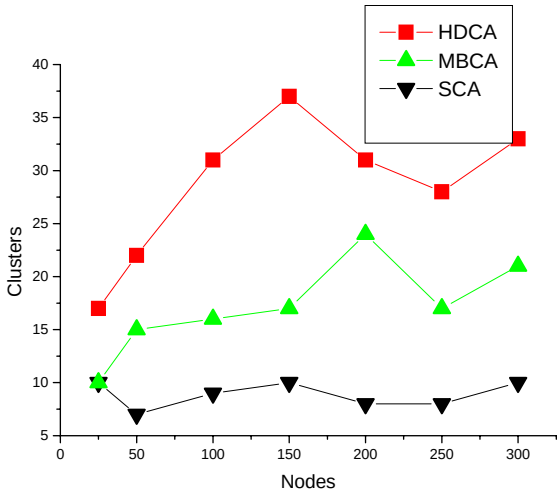
As shown in graph 1 and 2, the number of clusters created by the HDCA is very high to reach the threshold of 245 clusters in graph 1, compared to MBCA and SCA which create less number of clusters (150 cluster for 300 nodes in graph 1), because they create two hops clusters covering large area compared to HDCA which cover small area, having as radius the transmission range of the CH which is 20 m in the experiments of graph 1 and 100 m in the experimentation of graph 2, however the radius of the cluster created by MBCA and SCA is 40 m and 200 m respectively in graph 1 and graph 2, because they create two hops clusters.

Creating great number of clusters results on difficulties to manage and maintain the structure of clusters.

From these two experiments we can conclude that one hop cluster aren't suitable for large network deployed in a large area (500*500m or greater), because nodes are dispersed in this area and creates one or two nodes clusters which is too small, because the purpose of clustering is to manage great number of nodes in small groups, which isn't achieved using one hop clusters.



Graph IV.1Transmission range 20 m area 500*500 m



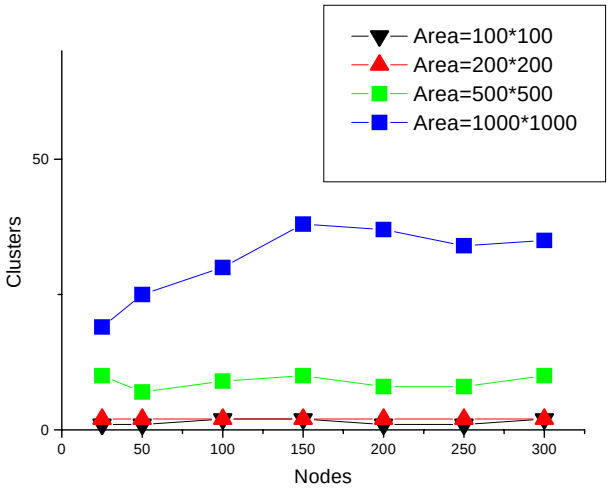
Graph IV.2 Transmission range 100 m Area 500*500 m

4-4 Proving the limitation of the number of CMs in each cluster

In this section we are going to analyse the reaction of HDCA and MBCA clustering algorithm in some scenarios to prove the utility of the upper bound proposed in SCA to limit the maximum number served simultaneously by a CH, because these two algorithms don't make any assumption on the maximum number of nodes supported by a CH.

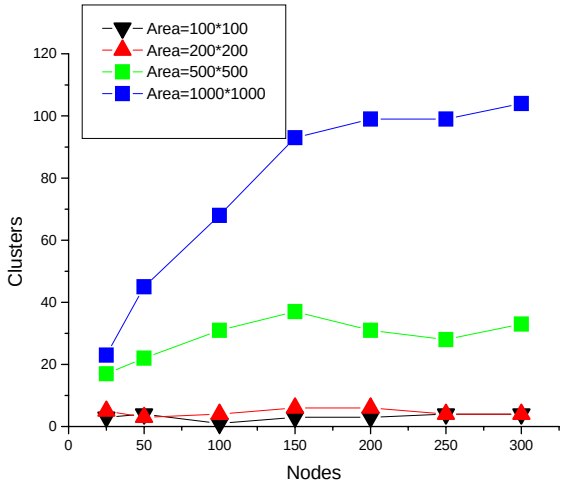
For the area of 1000*1000 m the number of cluster is always great for all algorithm, because nodes are out of the transmission range of each other. The SCA and MBCA manage this by the extension of the size of the cluster size from D to D+1 which may reduce the number of clusters.

In graph 3 are shown the experiments made on the MBCA where we can observe that for the transmission range of 100 m in small area 100*100 m, 200*200 m and 300*300 m, the algorithm create a very small number of clusters which stays less than 10 clusters in the area of 300*300 m and less than three in area 100*100 m and 200*200 m, which is too small to manage the increasing number of nodes because CHs are mobile nodes, which can't support and serve great number of nodes (150 nodes) simultaneously.



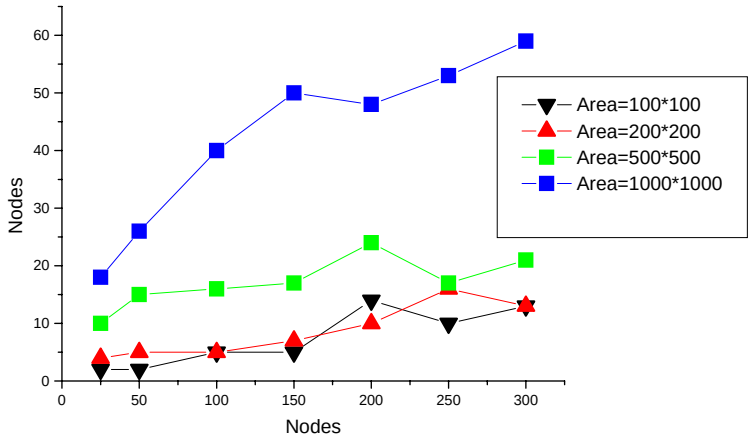
Graph IV.3 MBCA, transmission range=100

In graph 4 are shown the experiments made on the HDCA, as we can predict it has the same problem as the MBCA because it doesn't make any assumption on the maximum number nodes supported by a CH. Thus it creates small number of clusters to manage the increasing number of nodes. For example in the area of 100*100 and 200*200 it always create less than 5 clusters to manage a set of nodes going from 20 nodes to 300 nodes. However it creates more clusters for the areas of 300*300 m and 1000*1000 m because the nodes are out of the transmission range of each other.



Graph IV.4 HDCA, transmission range=100

In graph 5 are shown the experiments made on the SCA for different size of the area. As we can observe SCA manage the increasing number of nodes in the network by creating more clusters. This is done because the number of nodes in each cluster is limited, therefore when there is more nodes in the network the algorithm manage them by creating more clusters. For example for the area of 100*100m, 200*200 m and 300*300 m the average number of nodes in each cluster stay around 25 nodes per cluster which is reasonable considering the inputs of simulation. In opposition of HDCA and MBCA keep the same number of cluster for all network configurations.



Graph IV.5 SCA, transmission range 100 m, D=2

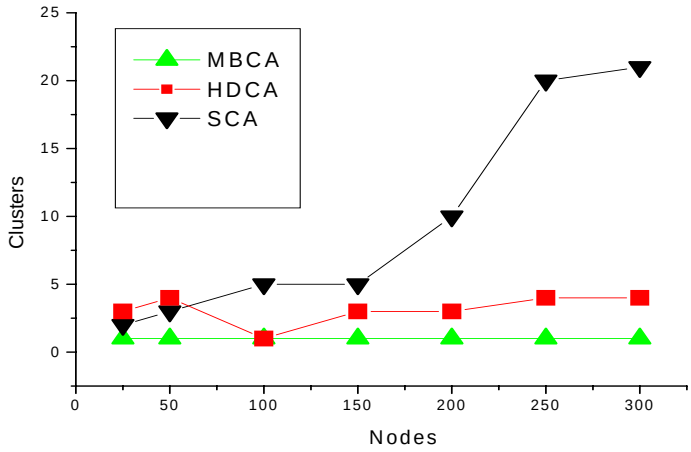
Conclusion

From the previous experiments we can conclude that an upper bound limiting the maximum number of nodes in each cluster is imposed by the increasing number of nodes and the area where the ad hoc network is deployed. The two algorithms analysed in this work don't treat this aspect so they create clusters with great number of nodes. Therefore in SCA we have proposed to limit the number of nodes in each cluster we have also proposed that this maximum boundary is defined according to the remainder of resources in each node serving as CH.

4-5 Test of performance of SCA for dense networks

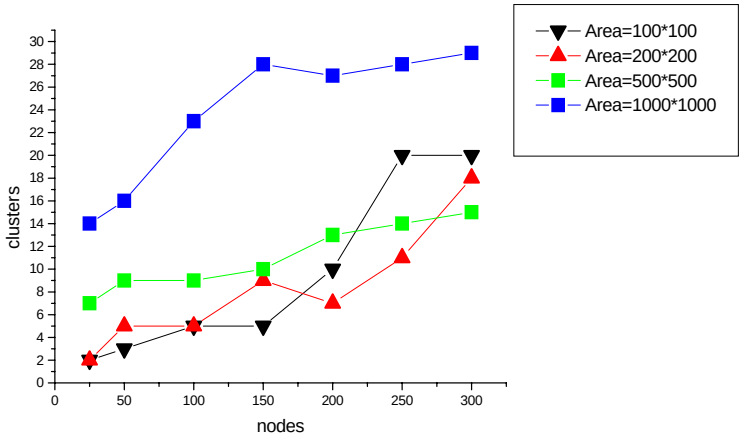
In this section we are going to test the performance of SCA in different situation compared to MBCA and HDCA.

In graph 5 we compare SCA with MBCA and HDCA we have also fixed the number of hops to 3. As we can see SCA manages the increasing number of nodes by creating more clusters, to reach the threshold of 21 clusters for 300 nodes, resulting on the average of 15 nodes per cluster which is reasonable in ad hoc network. However the MBCA and HDCA create always the same number of clusters for all the configuration of the network for example the MBCA create one cluster to manage 300 nodes which is impossible in ad hoc network. This is also the case for HDCA which creates less than 4 cluster to manage 300 nodes.



Graph IV.6 Transmission range 100 m Area 100*100m D=3

In graph 7 we are testing the reaction of SCA in different areas with the radius of 3 hops. Thus we observe that SCA manage always the increasing number of nodes in the network by creating more clusters to stay in the same average of nodes which is randomly chosen between 10 and 30 nodes. We can also observe that in large area (1000*1000 m) it creates big number of clusters like we can observe in graph 5, however to overcome this the size of clusters is augmented from 2 hops to 3 hops which results on the diminution of the number of clusters, this results is observed in graph 7 in which the number of clusters created in the area of 1000*1000 m doesn't reach the 30 clusters, however with the same configuration and the size is 2 hops it creates more than 55 clusters.



Graph IV.7 SCA, transmission range 100 m, D=3

5-Conclusion

In this chapter we've presented our contribution in the domain of clustering, in which we have described our proposed algorithm of clustering, in which we've proposed new mechanisms to overcome the limits encountered in other algorithms taken from the literature by:

- Including the aspect of security by proposing to use a voting mechanism to elect the most trusted node, we've also proposed to use certificate as identifier to avoid spoofing attacks.
- Defining a set of system parameters for the election procedure to elect the most power node as cluster-head. Including a new method to compute stability more simple and possible to be used in ad hoc network.
- Defining an upper and lower bound to limit the number of nodes in clusters, giving birth to new mechanism as cluster size extension and cluster size diminution.
- Creating D hops cluster, resulting on a less number of clusters which may also reduce the number of roaming requests, and management difficulties.
- We've proposed to use efficient flooding to broadcast beacons or any other request over the network, to avoid overhead caused by flooding.
- Finally we've proven by simulation some of the preceding choices. We have also compared the efficiency of our algorithm with other algorithm taken from the literature considering the number of nodes in clusters.

SCA algorithm is developed specially to serve security protocols based on clustering like the cluster based Public Key Infrastructure. However it can be used for other purposes like routing because SCA manage efficiently the clustering architecture. It also uses a set of criteria to stabilize the clustering architecture which is useful for routing protocol.

Another aspect making it possible to be used for different protocols and configurations is that the set of system parameters used in the election procedure and the size of the clusters are defined according to the need of the network or the underlying protocol.

Conclusion

In this modest work we've tried to study the problem of security in ad hoc networks. Considering the characteristics of ad hoc networks ranging from frequent topology changing to constraint on mobile nodes, it appears that the development of an effective mechanism of security still in laboratories studies.

In our thesis we've begun by giving a brief introduction to ad hoc networks and their problematic. Then we've given a state of the art of recent works to establish key management in ad hoc network. We've also proved that these solutions aren't directly applicable for ad hoc networks. In our thesis we've chosen to implement Public Key Infrastructure (PKI), since it's known as the most effective tool providing security in wired networks. However it isn't clear if it can be or not implemented in ad hoc network.

Therefore we've proposed our contribution in the domain of the adaptation of PKI for ad hoc network, employing clustering to manage the complexity of these infrastructure insuring in this way most of the criteria of successful key management scheme, compared to other PKI key management schemes it seem that our proposed solution is the most efficient and possible to be applicable in ad hoc networks, since it uses clustering to group the network and facilitating the management of certificate authority. We've also given a new method for authentication for node joining using location limited channels. We've also included the use of symmetric key encryption to encrypt ordinary traffic over the network accompanied with a proactive key update to enforce security by periodically change the encrypting key.

To make in practice our design we've given a prototype implementing all the operations defined to mange clustered PKI.

We've also proposed a new clustering algorithm, in which we've tried to include the aspect of security by using a trust value and other system parameters making it more efficient and effective compared to other protocols. The tests on this clustering algorithm are done using ClusterSIM developed especially for simulating clustering algorithms.

Finally the aspect of security still in development and new approaches and mechanisms appear every day, accompanied with a fast development of the use of ad hoc networks. It seems also that these networks are democratised and gain application in our life, therefore any protocol must be simple as possible to be used by no professional persons. We think that the use of clustering to manage PKI is more simple compared to other, since clustering is more evident for humans.

Annex 1

1- Description of ClusterSIM

ClusterSIM is a simulator developed during the elaboration of this thesis, to perform the needed simulation and performance analysis of clustering algorithms. The results given by ClusterSIM are the same as given in different publication as [60] on which we've focused our tests; therefore we've supposed the validity of this simulator.

ClusterSIM was implemented in JAVA using the standard edition J2SE. Using JAVA we've given to ClusterSIM more areas of use as Linux and Windows. It consists of a set of intuitive classes (mobile node, coordinate….) simple to understand and can be extended later for other purposes.

ClusterSIM was developed to simplify the implementation of clustering algorithms. In the way that ClusterSIM gives a set of intuitive methods can be changed for each algorithm depending on its specifications.

2- Class Diagram

In this section we are going to show the class diagram of ClusterSIM, regarding the difficulty of representing all the classes of the simulator only important classed are described in this section. Thus ClusterSIM is mainly composed of the following classes:

- Clustering_Algo: this class implements the election procedures of each algorithm; therefore to implement new algorithms we must only write the election procedure by considering the specifications of that algorithm.
- Area: this class is used to define the area where nodes are authorised to move.
- Coordinate: this class implements the structure of nodes coordinates (X and Y) and other parameters as the direction of the node. An instance of this class is attached to each instance of the class Node.
- Node : this class implements the most important methods that can be done by any mobile node in an ad hoc network as:
    - Init: this procedure initiates the parameters of the considered node such as speed, initial coordinates, direction and wait time.
    - Move: this method changes after each second the coordinate of the node according to its speed. Therefore after each second new value are given to X and Y, however if the value of X or Y reaches the boundaries of the considered area the direction of the node is changed.
    - Set_Cluster-head: this methods change the cluster-head of the considered node, it has as input parameter the reference of the cluster-head.
    - Distance: this method returns the distance between two nodes by considering its transmission range and the coordinates of each node.
    - Get_neighbours: this method returns the number of neighbour nodes according to the transmission range and the radius of clusters (one, two hops). It uses the coordinates of each node and compare them with its own coordinate to compute the distance between them and conclude if that node is neighbour or not.

- Graph_Plot and Animation: These classes are responsible of plotting the simulation process on the screen.
- Simulator: this class implements the main window of the user interface.
- Results, Statistics: these two classes are used to save simulation results.

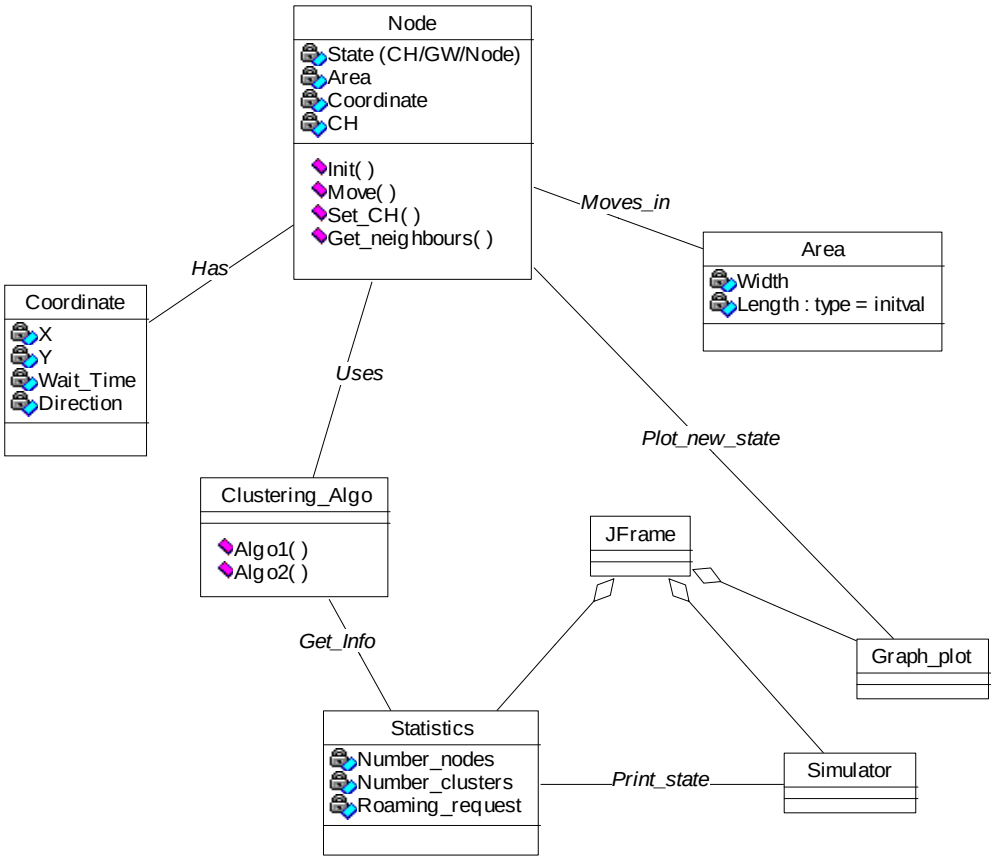


Figure A.1 Class Diagram

3- Description of the user interface

The user interface of ClusterSIM is very intuitive and simple since it doesn't contain lot of commands.

The main window contains a main menu and a tool bar containing the most important commands for simulation.

1. Creating new simulation: using this button the user can inter the parameters of new simulations, as shown in Figure 2 the user must inter the max speed, transmission range, wait time, number of nodes, the size of the area and simulation time.

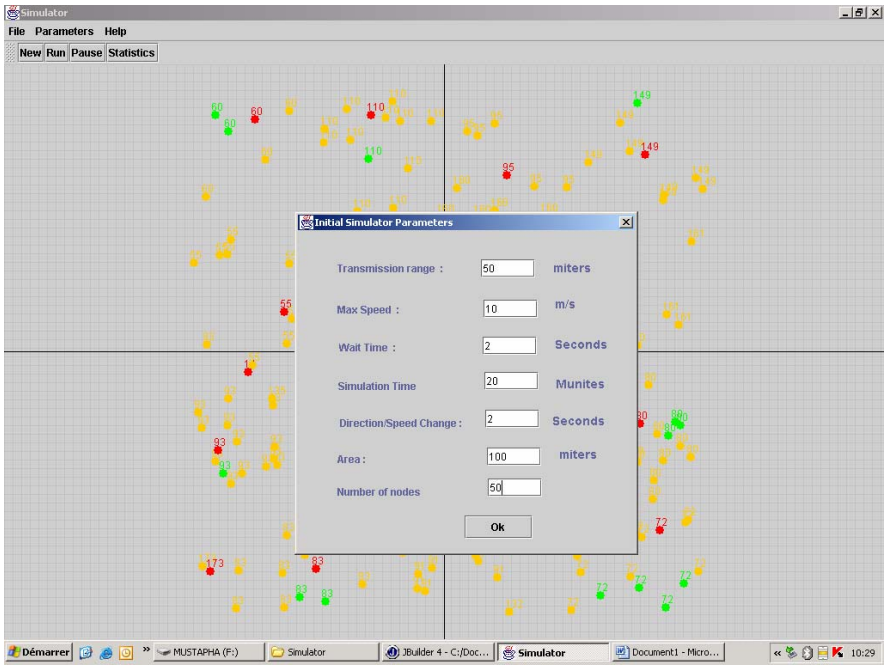


Figure A.2 Parameters introducing

2. Running the simulation: this command is launched using the button run, it can be used only if the simulations parameters are interred using the simulation creation command.
3. Viewing statistics: after the achievement of the simulation the results about the simulation are automatically plotted on screen Figure 3, however the user can view these results during the simulation using the statistics button.

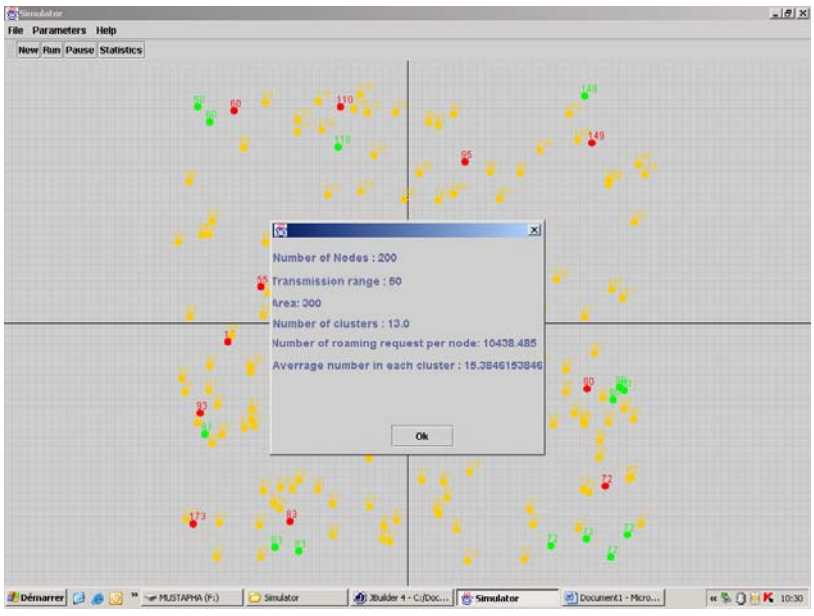


Figure A.3 Statistics Show

The simulation process and nodes movement can be viewed on the screen for each simulation giving the user more comprehension about the current simulation.

The nodes are represented in the form of a circle moving randomly in a limited area, to differentiate them each kind of nodes, was given a colours:

- Cluster-head is presented using red colour.
- Cluster-members (ordinary nodes) are represented with green colour.
- Gateways are given orange colour.
- No determinist nodes are given blue colour.

4- Implementing new algorithms

To implement new algorithms two major modifications are needed:

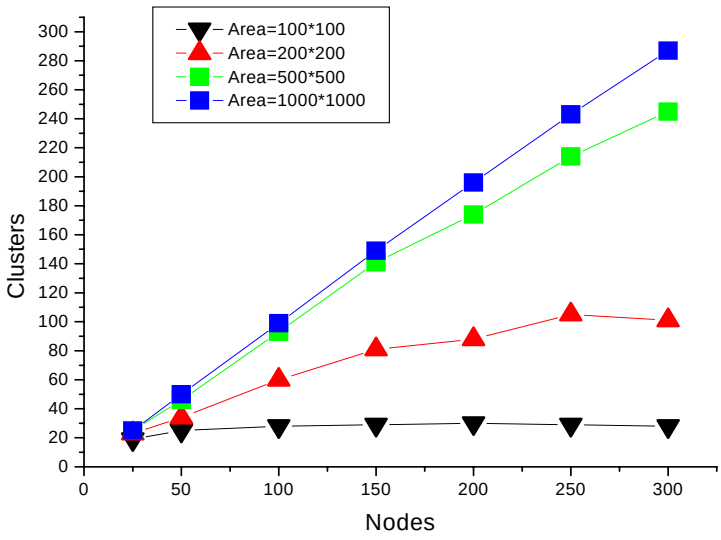
- Writing new election procedure: this step consists of analysing other implemented election procedure and according to the specification of the new algorithm to implement the election procedure. This new procedure must benefits from node's class methods to achieve the election procedure. Otherwise if the new algorithm needs more parameters, new methods must be implemented on the Node class. For example when the clustering algorithm is energy aware it must implement a new method to modify the battery state of the node.
- Adding the new procedure: after having writing the election procedure, we must add a line in the maintenance procedure. The maintenance procedure is executed every second and chooses the appropriate election procedure and executes it, therefore new test is added for each new algorithm.

```
if(sim.algo.equals("DHop")) dhop();
if(sim.algo.equals("SecuredA")) SecuredA();
```

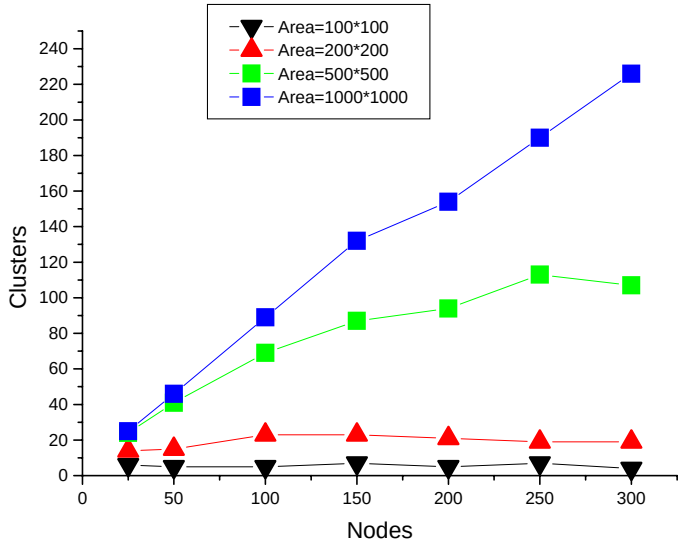
- Now the new algorithm is ready to be executed.

Annex 2

1- Highest degree clustering algorithm

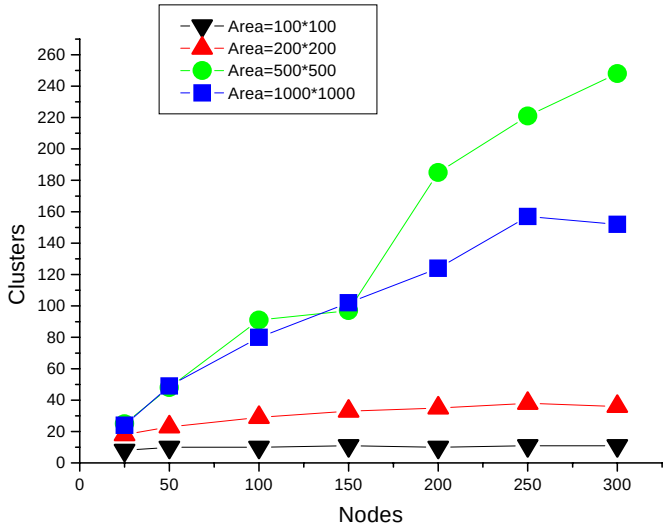


Graph A.1 Test of HDCA in different areas
Transmission range 20 m

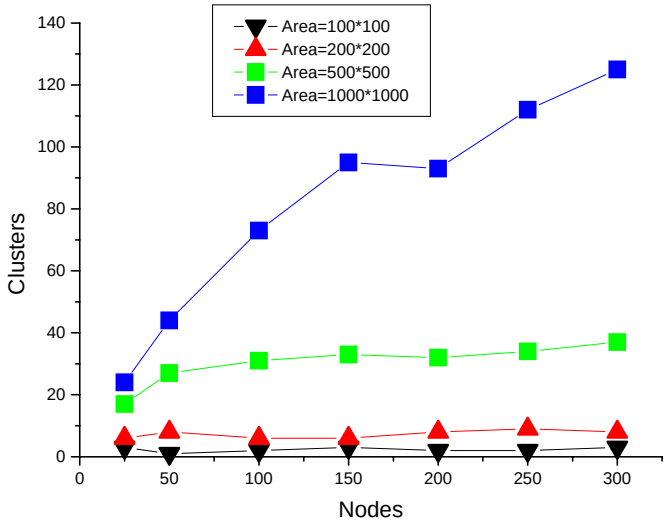


Graph A.2 Test of HDCA in different areas
Transmission range 50 m

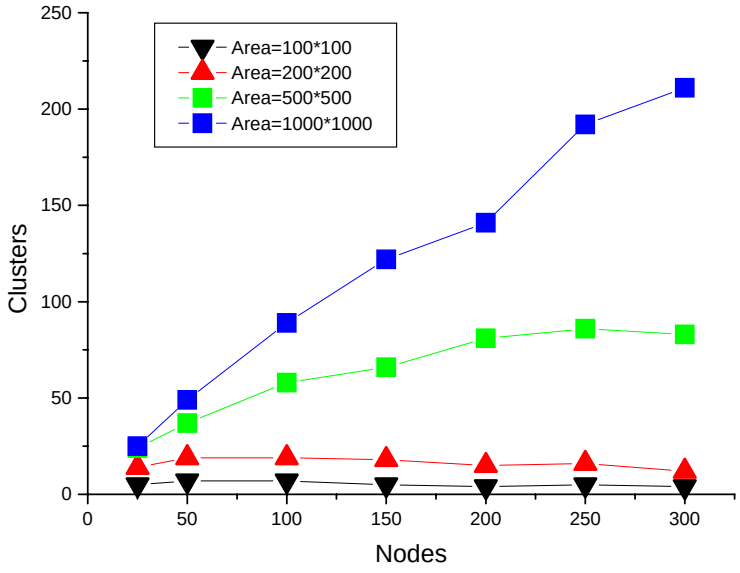
2- Mobility based clustering algorithm



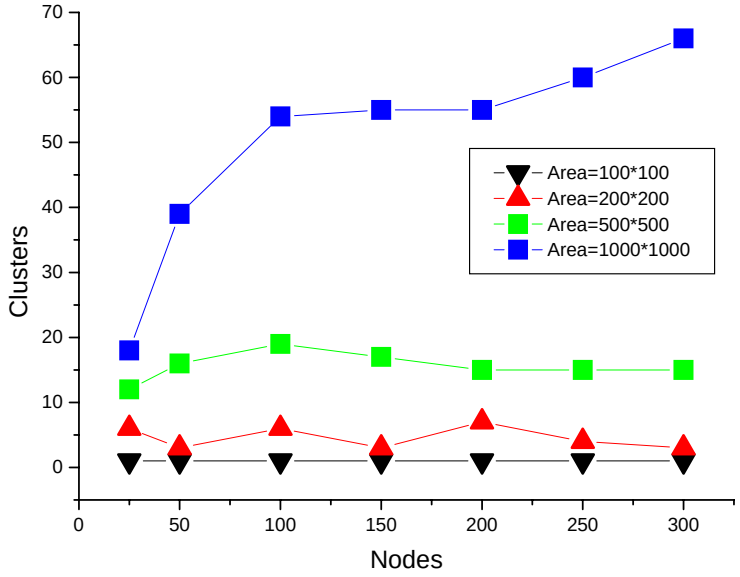
Graph A.3 Test of MBCA in different areas
Transmission range 20 m D=2



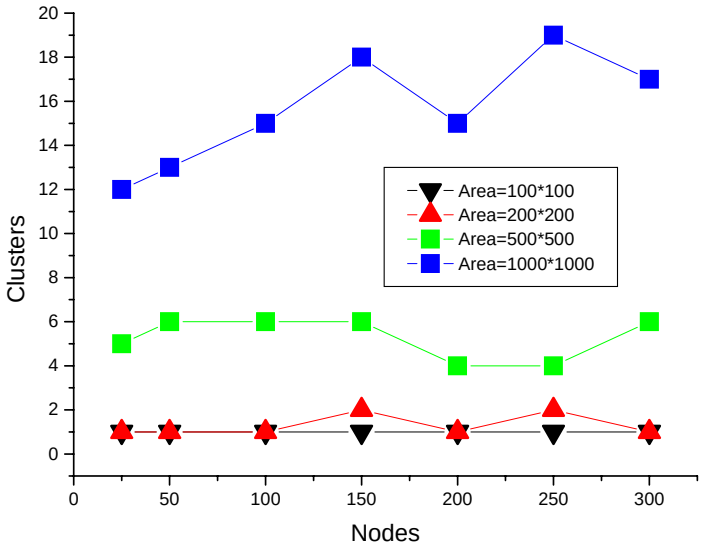
Graph A.4 Test of MBCA in different areas
Transmission range 50 m D=2



Graph A.5 Test of MBCA in different areas
Transmission range 20 m D=3

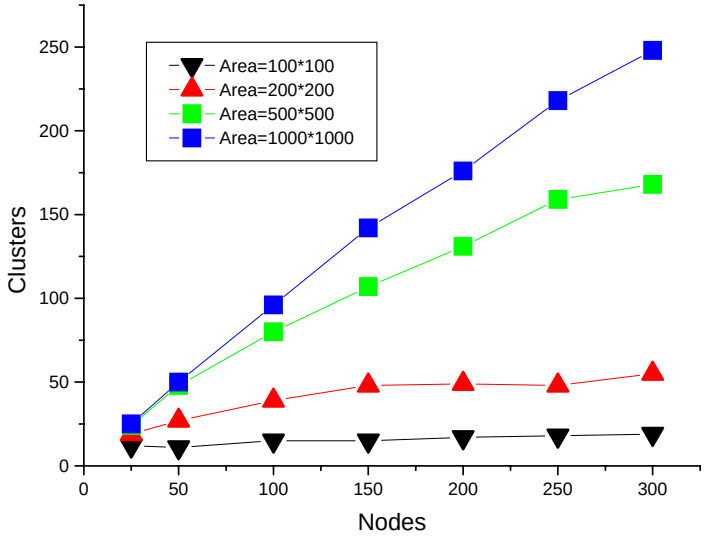


Graph A.6 Test of MBCA in different areas
Transmission range 50 m D=3

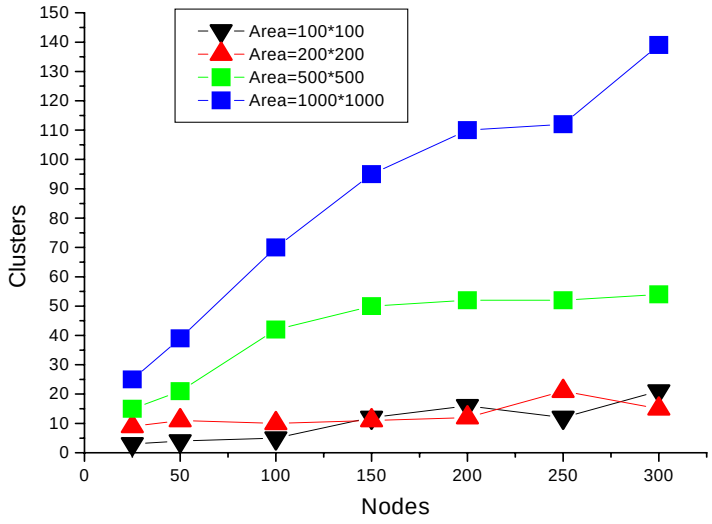


Graph A.7 Test of MBCA in different areas
Transmission range 100 m D=3

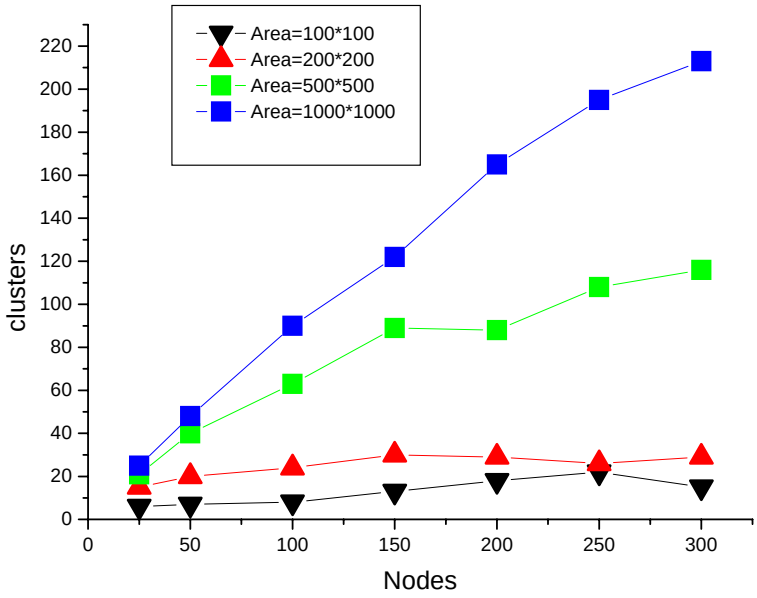
3- Secured clustering algorithm graphs



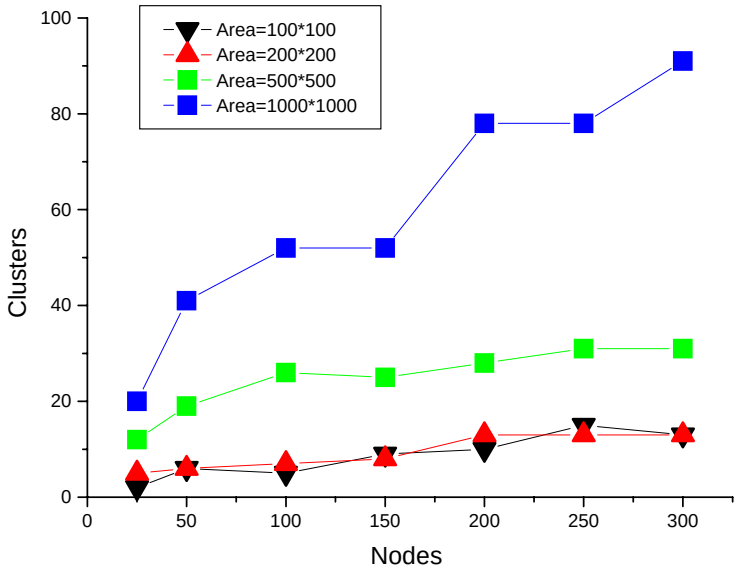
Graph A.8 Test of SCA in different areas
Transmission range 20 m D=2



Graph A.9 Test of MBCA in different areas
Transmission range 50 m D=2



Graph A.10 Test of SCA in different areas
Transmission range 20 m D=3



Graph A.11 Test of SCA in different areas
Transmission range 50 m D=3

References

[1]Nicolas Prigent, Christophe Bidan, Olivier Heen, and Alain Durand. *Securite des réseaux domestiques : optimaux les grands remèdes*. Actes du symposium SSTIC03

[2]Guy Pujolle. Securité Wifi. Eyrolles, 2004.

[3]NIST (National Institute of Security and Technologies). *Wireless Network Security 802.11, Bluetooth and Handheld Devices*. Technical report

[4]Shariful Islam. *Efficient Key Management Scheme for Mobile Ad Hoc Network*. Master of Science. Royal Institute of Technology (KTH) SecLab Department of Computer and System Sciences (DSV). Stockholm, Sweden, 2005.

[5]Klas Fokine. *Key Management in Ad Hoc Networks.* Linköping University Electronic 2002.

[6]Paul Mühlethaler, *Security Schemes for the OLSR Protocol for Ad Hoc Networks*, doctoral thesis. University paris 6 – pierre et marie curie, 2005.

[7]I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. *Cayirci Wireless sensor networks: a survey.* Computer Networks 38 (2002) 393–422.

[8]Ossama Younis and Sonia Fahmy. *Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach.* IEEE 2004.

[9]Fei Hu, Jason Tillett, Jim Ziobro, Neeraj K. Sharma. *An Energy-efficient Approach to Securing Treezone-based Sensor Networks*. GLOBECOM 2003, pp 1430, 1434.

[10]Patroklos g. Argyroudis and Donal O'mahony. *Secure Routing For Mobile Ad Hoc Networks*. IEEE Communications Surveys & Tutorials • Third Quarter 2005.

[11]D. B. Johnson and D. A. Maltz, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, IETF Internet draft, 19 July 2004, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt

[12]C. E. Perkins et al. *Ad hoc On-Demand Distance Vector (AODV) Routing,* RFC 3561, July 2003, http://www.ietf.org/rfc/rfc3561.txt

[13]T. Clausen, and P. Jacquet, *Optimized Link State Routing Protocol (OLSR),* RFC 3626, October 2003, http://ietf.org/rfc/rfc3626.txt

[14]Haas, Z.J., Pearlman, M.R. and Samar, P. *Zone Routing Protocol(ZRP).* IETF Internet Draft, draft-ietf-manet-zrp-04.txt, July 2002.

[15]Mingliang Jiang, Jinyang Li, Yong Chiang Tay. *Cluster Based Routing Protocol(CBRP) Functional Specification.* IETF Internet Draft, draft-ietf-manet-cbrp-spec-00.txt August 1998.

[16]Panagiotis Papadimitratos, Member, IEEE, and Zygmunt J. Haas, Senior Member, IEEE. *Secure Data Communication in. Mobile Ad Hoc Networks*. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 2,2006 pp 343,356.

[17]Mohsen Guizani. "*Security and Trust in Mobile Ad Hoc Networks"*. Proceedings of the 4th Annual Communication Networks and Services Research Conference (CNSR'06). 2006.

[18]konrad Wrona. *"Distributed security: ad hoc networks & beyond"*. Ad Hoc networks security, pompas workshop, 2002.

[19]Mohamad Badra. *Le transport et la sécurisation des échanges sur les réseaux sans fil.* Doctorat thesis, Ecole Nationale Supérieure des Télécommunications France 2004.

[20]Kamil Kulesza, Zbigniew Kotulski, *Countermeasures against traffic analysis for open networks*, Institute of Fundamental Technological Research, Polish Academy of Sciences, Warsaw Poland. ENIGMA 2005.

[21]Adam Burg, *Ad hoc networking: concepts, applications, and security*. Ad hoc network specific attacks Seminar, Technische Universität München, 2003.

[22]Tor Inge Skaar, Tor-Erik Thorjussen, *Security Specification, Access Control and Dynamic Routing for Ad-Hoc Wireless Networks applied to Medical Emergencies*. Project report Norwegian University of Science and Technology Faculty of Information Technology, Mathematics and Electrical Engineering, 2003.

[23]Souheila Bouam , Jalel Ben-Othman. *Protocole de Sécurisation des Données à Base de Routage Dans les Réseaux Ad hoc*. Laboratoire CNRS-PRiSM, Université de Versailles, France.

[24]Chang Yuan-Yao. *802.11 Person-In-Middle (PiM) Attacks: Implementation and Practical Solutions*. A Thesis Presented to The Faculty of the School of Engineering and Applied Science University of Virginia, 2004.

[25]Tara M., Charles R.Elden,2002. *Wireless security and privacy Best Practices and Design Techniques*, Addison Wesley.

[26]Stinson Douglas, Vande May Serge. Cryptographie : théorie et pratique. Vuibert, 2001.

[27]John Rittinghouse, James Ransome, 2004. *Wireless operational security*, Digital Press.

[28]Seung Yi Robin Kravets, *Practical PKI for Ad Hoc Wireless Networks*, Department of Computer Science University of Illinois at Urbana-Champaign, *Report No. UIUCDCS-R-2002-2273, UILU-ENG-2002-1717,*August, 2001.

[29] Seung Yi, Robin Kravets. MOCA : Mobile Certificate Authority for Wireless Ad Hoc Networks.

[30]L. Zhou and Z. J. Haas, *Securing Ad Hoc Networks*. IEEE Networks, Volume 13, Issue 6 1999.

[31]M. Bechler., H.-J. Hof, D. Kraft, F. Pahlke, L. Wolf. A Cluster-Based Security Architecture for Ad Hoc Networks.

[32]Mohamed Elhoucine Elhdhili, Lamia Ben Azzouz, Farouk Kamoun. A Totally Distributed Cluster Based Key Management Model for Ad hoc Networks.

[33]H. Luo and S. Lu, 2000. *Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks.* Technical Report 200030, UCLA Computer Science Department.

[34]J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, 2001. *Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks.* IEEE ICNP.

[35]H. Luo, P. Zerfos, J. Kong, S. Lu and L. Zhang, 2002. *Self-securing Ad Hoc Wireless Networks.* IEEE ISCC.

[36]J-P. Hubaux, L. Buttyán and S. Capkun, 2001. *The Quest for Security in Mobile Ad Hoc Networks.* ACM.

[37]Network Associates, Inc. and its Affiliated Companies. 1998. Une Introduction à la Cryptographie, PGP, Version 6.0.2

[38]Kamran Jamshaid, Loren schwiebert , Secure and Efficient Key Exchange for Sensor Networks, IEEE 2004.

[39]S. Basagni, K. Herrin, E. Rosti and Danilo Bruschi, 2001. *Secure Pebblenets.* ACM.

[40]D. Balfanz, D. K. Smetters, P. Stewart and H. Chi Wong, 2002. *Talking To Strangers: Authentication in Ad-Hoc Wireless Networks.* Internet Society, Conference Proceeding of NDSS Conference.

[41]Lakshminarayanan. TAP - PRACTICAL SECURITY PROTOCOLS FOR WIRELESS PERSONAL DEVICES A. IEEE 2004.

[42]N. Asokan, P. Ginzborg, 2000. *Key Agreement in Ad Hoc Networks.* Computer Communications.

[43]MAINAK CHATTERJEE, SAJAL K. DAS and DAMLA TURGUT. *WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks.* Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204.

[44]M. Gerla and J.T.C. Tsai. *Multicluster, mobile, multimedia radio network, Wireless Networks.* 1(3) (1995) 255–265.

[45]Stephen Northcutt, Judy Novak, Donald McLachlan. *Detection des intrusion réseaux*. Compuspress, 2001.

[46]Cay S.Horstmann, Gary Cornell. Au Coeur de JAVA Notions fondamentales Volume I et II. CAMPUSPRESS, 2001.

[47]RMI Client Application Programming Interface, Java Card™ 2.2, Java™ 2 Platform, Micro Edition. Sun Microsystems, Inc. June 2002.

[48]Programmer's Guide J2ME™ Personal Basis Profile, Version 1.0 Java™ 2 Platform, Micro Edition. Sun Microsystems, Inc. June 2002 Revised July 10, 2002.

[49]Colette Johnen, Le Huy Nguyen. *Self-Stabilizing Clustering Algorithm for Ad hoc Networks*, technical report, LRI–Université Paris Sud,18 janvier 2006.

[50]Tomas Johansson and Lenka Carr-Motyckova. *On Clustering in Ad Hoc Networks*. Division of Computer Science and Networking, Lulea University of Technology, August 17, 2003.

[51]Javesh Boodnah and Eric M. Scharf. *Applying Clustering to a Framework for Generating Trust.* Queen Mary, University of London.

[52]Stefano Basagni, Michele Mastrogiovanni, Alessandro Panconesi, and Chiara Petrioli. *Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison*. IEEE Transactions on parallel and distributed systems, vol. 17, no. 4, 2006, pp 292,306.

[53]Mario Gerla. *Clustering and Routing in Large Ad Hoc Wireless Nets.* Computer Science Department University of California, Los Angeles, California 90095 Final Report 1998-99 for MICRO project 98-044.

[54]Farid Jaddi,Béatrice Paillase. *A Cluster Procedure for the Dynamic Source Routing Protocol in Ad hoc Networks.*, Med-Hoc-Net, The Third Annual Mediterranean Ad Hoc Networking Workshop, 2004.

[55]An Huiyao, Lu Xicheng, and Peng Wei. *A Cluster-Based Multipath Routing for MANET*. Med-Hoc-Net, The Third Annual Mediterranean Ad Hoc Networking Workshop, 2004.

[56]Yongcai Wang, Qianchuan Zhao and Dazhong Zheng. *Energy-Driven Adaptive Clustering Data Collection Protocol in Wireless Sensor Networks.* Proceedings of the

International Conference on Intelligent Mechatronics and Automation Chengdu,China August 2004.

[57]Yunjung Yi, Mario Gerla, Taek-Jin Kwon. *Efficient Flooding in Ad-Hoc Networks using On-Demand (passive) Cluster Formation.* Proceedings of Mobihoc, Jun 2003.

[58]Tzu-Chiang Chiang, Po-Yi Wu and Yueh-Min Huang. *A Limited Flooding Scheme for Mobile Ad Hoc Networks.* IEEE 2005.

[59]D.J. Baker and A. Ephremides. *The architectural organization of a mobile radio network via a distributed algorithm.* IEEE Transactions on Communications COM-29 11 (1981) 1694–1701.

[60]I.I. ER, and Winston K. G. SEAH. *Mobility-based D-hop Clustering Algorithm for Mobile Ad hoc Networks.* IEEE WCNC, Atlanta, USA, March 2004.

[61]Tsung-Chuan Huang, Liang-Cheng Shiu, Han-Chun Ke. A double manager k-hop clustering algorithm in mobile ad hoc networks. IEEE, 2004.