8

Trust Establishment in Mobile Ad Hoc Networks: Key Management

Dawoud D.S.[1], Richard L. Gordon[2],
Ashraph Suliman[1] and Kasmir Raja S.V.[3]
[1]National University of Rwanda
[2]University of KwaZulu Natal
[3]SRM University, Chennai,
[1]Rwanda
[2]South Africa
[3]India

1. Introduction

Mobile ad hoc networks are complex wireless networks, which have little or no existing network infrastructure. These networks can be established in a spontaneous manner allowing organizations and network members to work together and communicate, without a fixed communication structure. The mobility, spontaneity and ad hoc nature of these networks makes them optimal solutions for disaster area communication and tactical military networks. Due to recent wireless technology advances, mobile devices are equipped with sufficient resources to realize implementation of these dynamic communication networks. However, for ad hoc networks to find a wide spread within both the military and commercial world, they must be secured against malicious attackers.

Mobile ad hoc networks have distinct characteristics, which make them very difficult to secure. Such characteristics include: the lack of network infrastructure; no pre-existing relationships; unreliable multi-hop communication channels; resource limitation; and node mobility. Users cannot rely on an outside central authority, like a trusted third party (TTP) or certificate authority (CA), to perform security and network tasks. The responsibility of networking and security is distributed among the network participants. Users have no prior relationship with each other and do not share a common encryption key. Therefore, only after the network has been formed, the users establish trust and networking links. The establishment of networking links is identified as being vulnerable to security attacks. Trust establishment should allow protection for the network layer and ensure that honest links are created.

The sporadic connectivity of the wireless links, inherent to mobile ad hoc networks, results in frequent link breakages. These characteristics introduce unique challenges to trust establishment. Both the routing and trust establishment protocols must be designed to handle the unreliable wireless communication channels: the dynamic topology changes and the distributive nature. The security solutions used for conventional wired networks cannot simply be applied to mobile ad hoc networks. More complex network management must be implemented to achieve trust establishment in mobile ad hoc networks.

Ad hoc network security research initially focused on secure routing protocols. All routing schemes however, neglect the crucial task of secure key management and assume pre-existence and pre-sharing of secret and/or private/public key pairs [Zhou & Haas, 1999]. This left key management considerations in the ad hoc network security field as an open research area. Security solutions which use cryptographic techniques rely on proper key management to establish trust. This chapter together with the next chapter focus upon key management which aids these cryptographic solutions.

Outlines of the Chapter

This chapter and the next chapter form one unit. The two chapters focus largely upon establishing trust in mobile ad hoc networks, and concentrate more specifically on secure key management on the network layer. Our research focuses upon providing a solution for the security issues found in mobile ad hoc networks.

The current chapter is organised in the following manner: Section-2 provides a theoretical background to mobile ad hoc networks and the security issues that are related to such networks. These networks and their characteristics are defined in terms of trust establishment. As the focus of this research is on the network layer, attacks specific to this layer are identified and explained.

Section-3 presents a survey of the existing key management solutions for mobile ad hoc networks. Discussions are based on: functionality; availability; security services; scalability; efficiency; and computational cost. A comparative summary is presented, which identifies the difference in the requirements and the application of each solution.

In the next chapter , Section-2, we continue the discussions given in Section-3 of this chapter by offering a survey of the existing secure routing protocols for mobile ad hoc networks. The two sections identify the problem that the two chapters are addressing. There exists secure routing mechanisms to address the unique characteristics of mobile ad hoc networks, however, these solutions assume that key management is addressed prior to network establishment. A novel, on-demand solution to the key management problem for mobile ad hoc networks will be introduced in next chapter. The implementation of the proposed model, simulation of the model, the results and there analysis are given in next chapter.

2. Mobile ad hoc networks

An ad hoc network is a network with no fixed infrastructure. It allows for users to enter and exit any time, while seamlessly maintaining communication between other nodes. Mobile Ad Hoc Networks (MANETs) are advanced wireless communication networks which operate in an ad hoc manner. The term ad hoc is defined as:

"Meaning "to this" in Latin, it refers to dealing with special situations as they occur rather than functions that are repeated on a regular basis." (The American Heritage Dictionary of the English Language, Fourth Edition. Houghton Mifflin Company, 2004)

This definition suggests that it is a network which is formed in a spontaneous manner so as to solve an immediate communication need between mobile nodes. Mobile ad hoc networks differ from existing wired networks because they do not rely on a fixed network infrastructure [Capkun et al., 2003] [Haas et al., 2002], such as base stations or mobile switching centres. Instead, network functionality (e.g., routing, mobility management, etc.) is adopted by the nodes themselves. When using a multi-hoping routing protocol, mobile nodes within each other's radio range communicate directly via wireless links. However the nodes that are far apart depend on the other nodes to relay the message in a multi-hop fashion. Figure 1 [Zhou & Hass, 1999] demonstrates these autonomous, multi-hop characteristics. Connection between nodes is made by means of other nodes within the network. In Figure 1, the circle represents wireless range of node A. In Figure 1, when node D appears within the range of node A, the topology changes to maintain the connection. Note that all network functions are performed by the nodes and no host or outside authority exists.

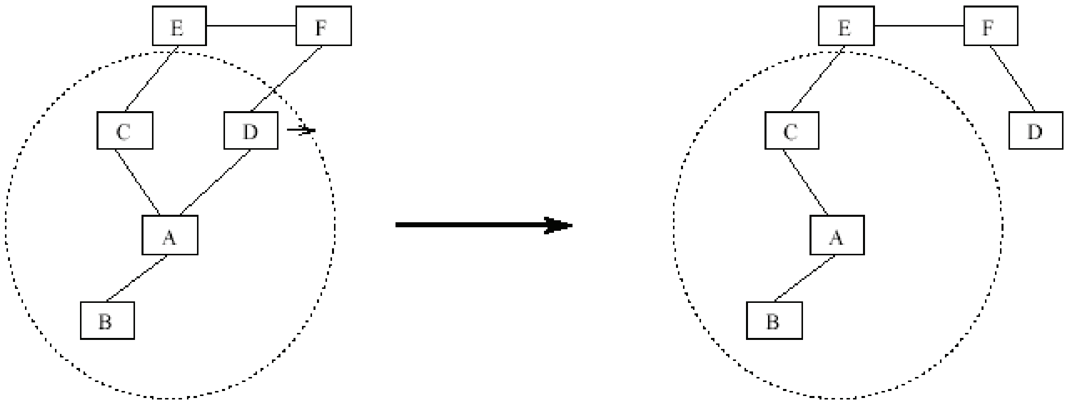


Fig. 1. Ad Hoc Network Topology

2.1 Application

Mobile ad hoc networks have become widely desired in military and commercial applications, due to the ever increasing development of mobile technology. The network's lack of infrastructure and independent nature allows for a robust network to be created within an unlikely networking environment.

a. Military Application

The first ad hoc networks were primarily deployed in the military domain in the early 1970's by the US Department of Defence, under the projects of DARPA and Packet Radio Network (PRnet) [Haas et al., 2002]. Ad hoc networks remain an important part of current and future military communication. They feature prominently in the following areas of military application: sensor networks; tactical networks; and positional systems.

Their application within the military field is based on the network's high mobility, survivability, and self-organized nature. This allows mobile military units to communicate effortlessly irrespective of the distance between each detachment. In a hostile environment, such as the battle field, an ad hoc network's distributive architecture eliminates the problem of a vulnerable network host or the loss of the network host. The modern battle field is characterized by highly mobile forces and the effect of a network which fails to maintain communication and high mobility is disastrous. An example of this can be seen in the experience of the Iraqi forces during the 1991 Gulf War. For this reason, soldiers would prefer mobile ad hoc networks, as opposed to existing local networks. Both invading and defending soldiers would avoid using the local operator, therefore ensuring communication stealth required for battle. Another illustration of the downfall of using an existing local network can be seen in Chechnya, where a general was killed by a missile which tracked the uplink signal of his portable phone. It is clear from these examples that mobile ad hoc networks provide stealth, mobility, and security in the battle field.

The military context is the most obvious application for mobile ad hoc networks. More recently in July 2008, DARPA invested $8.5 million in the Intrinsically Assurable Mobile Ad Hoc Network program (IAMANET) [Jameson, 2008]. This project aims to improve the integrity, availability, reliability, confidentiality, safety, non-repudiation of MANET communication and data in the future.

b. Commercial Application

Early application and developments were military focused. However, non-military applications have grown rapidly due to the availability and advances in mobile ad hoc research. The introduction of new standards such as IEEE 802.16e, IEEE 802.11g and IEEE 802.15.4, have significantly helped the deployment of wireless ad hoc network technology in the commercial domain [Haas et al., 2002]. In this sector the aforementioned networks are desirable due to their dynamic and self organized nature, which allows rapid network deployment. This is particularly useful in situations where infrastructure is damaged or does not exist, and where existing conventional networks are unaffordable or lack sufficient network coverage and need to be side-stepped. Some examples of these applications include: personal area networks; sensor networks; emergency networks; and vehicular communication

Personal area networks are created when a small number of nodes meet spontaneously to form a network for the purpose of teleconferencing, file sharing, or peer-to-peer communication. An example of this can be seen when attendees in a conference room share data using laptops or handheld devices.

Sensor networks are used to monitor data across an area. An example of these networks includes small sensor devices which are located in animals and other strategic locations that collectively monitor and analyze the environmental conditions. Sensor networks have also been developed, by the PermaSense Project, to monitor the permafrost found in the Swiss Alps [Talzi et al., 2007].

The application of this network to an emergency context often occurs in a hostile environment, similar to the military context. Natural or man-made disasters may result in the existing network infrastructure being unavailable or unreliable. Ad hoc emergency services could allow communication and sharing of video updates of specific locations, among relief workers and the command centre. An illustration can be seen in the event of the New York World Trade Centre disaster, on September 11, 2001. The majority of the phone base stations were knocked out in less than twenty minutes, after the attack. The remaining base stations were unable to operate because they could not work in ad hoc mode. The Wireless Emergency Rescue Team recommended afterwards that telecom operators provide ad hoc mode for their infrastructure in the event of emergency situations to enable co-operation between police, firemen and hospital networks [Karl & Rauscher, 2001]. Mobile ad hoc networks can allow for rapid network deployment in an emergency situation. Emergency networks can be set up in remote or hostile areas where there is no existing communication infrastructure, thereby assisting relief work and rescue missions.

A Vehicular ad hoc network provides communication between vehicles, roadside equipment and vehicles travelling in close proximity. Data is exchanged between nearby vehicles to provide traffic information and early warnings for accidents and road works. The purpose of Vehicular ad hoc networks is to provide a communication network of safety and information for users [Raya & Hubaux, 2005].

The benefits of ad hoc networks have realized new non-military communication opportunities for the public. Companies are starting to recognize the potential for commercial ad hoc network applications, and as a result laptops and handheld devices are being equipped with wireless functionalities. Businesses are offering products using ad hoc networking technology in areas of: law enforcement; intelligent transport systems; and community networking. These dynamic networks have still not reached their full potential, and it is clear that ad hoc technology has an imminent role to play in the development commercial technology of today and the future.

2.2 Ad hoc network challenges

An ad hoc network is a dynamic type of network which is both similar and very different to its parent fixed communication network. In the following we introduce the properties of an ad hoc network as a way of defining its shortcomings and to highlight its security challenges.

a. Dynamic Network Architecture

Ad hoc networks have no fixed or existing network infrastructure. The network architecture is continuously changing as the network evolves. There is no pre-existing or fixed architecture which handles all network tasks such as: routing security and network management. Instead, the network infrastructure is spontaneously set up in a distributive manner. Each participating node shares the network's responsibilities. Distribution of network functionality avoids single point attacks and allows for the network to survive under harsh network circumstances.

A fixed entity structure, such as a base station or central administration, is crucial for security mechanisms. A trusted third party member [William, 1999], which is expected in traditional networks, is similar to a fixed entity as both define security services; manage and distribute secret keying information (which allows secure communication of data through encryption and decryption techniques). Therefore the absence of such a control entity introduces new opportunities for security attacks on the network.

b. Self Organized Nature

Wireless ad hoc nodes cannot rely on an off-line trusted third party member. The security functions of the trusted third party member are distributed among the participating nodes. Each node takes responsibility for establishing and maintaining its own security and is, therefore, the centre of its own world and authority. A wireless ad hoc network is therefore referred to as a self organized network [Capkun et al, 2003].

c. No Prior relationships

In ad hoc networks, nodes can have no prior relationships with other nodes within the network. Prior acquaintance between nodes can be considered as pre-trust relationships between nodes. However, the ad hoc nature of these networks does not allow for these assumptions, as it cannot be assumed that secrets exist between the respective pair of nodes [Eschenauer & Gligor, 2002]. If nodes can join and leave the network at random without prior trust relationships with nodes, access control becomes a difficult task for the security mechanism.

d. Multi-hop communication channel

Wired networks include fixed nodes and fixed wired communication lines. Wireless ad hoc networks have mobile wireless nodes (often in the form of hand held devices) and, as suggested, their communication medium is wireless. This allows for greater network availability and easy network deployment. Each node's transmission range is limited and network communication is realized through multi-hop paths. Co-operation and trust along these paths is a crucial aspect of the security mechanism and ensures successful communication. The shared wireless communication medium means that any user can participate in the network. This creates access control problems for security mechanisms as adversaries are able eavesdrop on communication or launch active attacks to alter message data.

e. Mobility

Nodes are expected to be mobile within an ad hoc network, creating a dynamic and unpredictable network environment. In certain situations the nodes' mobility is not totally unsystematic and assumptions can be made in the form of mobility patterns [Capkun et al, 2006]. An example of these patterns is evident in a vehicular ad hoc network where vehicles move along fixed paths, or roads, at speeds which have a high probability of being within the local speed limit. However, nodes demonstrate random mobility within these predictions [Capkun et al, 2006].

Connectivity between nodes is sporadic. This is due to the shared, error-prone wireless medium and frequent route failures which caused by the unpredictable mobility of nodes [Van der Merwe & Dawoud, 2005]. Increased mobility can result in the multi-hop communication paths being broken and network services becoming unavailable. Security mechanisms must account for the weak connectivity and unavailability. Furthermore, due to mobility and sporadic connectivity, these mechanisms must also aim to be scalable with the changing network density.

f. Resource Limitations

Wireless nodes allow for the freedom of mobility and easy network establishment and deployment. Wireless nodes are often smaller hand-held devices that do not experience the same resource privileges of traditional wired nodes [Hass et al, 2002]. Mobile nodes are ideally low cost and small in size as to maximize node availability and mobility. In attempt to achieve these objectives wireless nodes have limited resource, specifically in the following areas:

- Battery life
- Communication range
- Bandwidth
- Computational capacity
- Memory resources

If mobility is to be attained, nodes must be battery powered. Battery powered nodes suffer from the consequences of power failures which break connectivity. They also run a high possibility of failing to be on-line the entire duration of the network. This could hinder network service availability. Cost and power restrictions limit the design features of wireless nodes. Power and transmission range are directly related, resulting in wireless devices having limited transmission ranges and bandwidths. Low powered, low cost CPU's are preferred, as this reduces the computational capacity and memory resources available for routing and security operations. As discussed above, network and security tasks are not performed by a central authority, but rather distributed among all the nodes. This creates a heavy burden upon the nodes to perform their own tasks as well as the network services. If the security mechanisms do not distribute the load fairly, adversaries can act in a selfish manner, forcing other nodes to perform extra tasks. In some instances malicious nodes will flood a single node with service requests in the aim of depleting its limited resources. A well designed security algorithms optimizes computational processing and operation to meet the limited resource requirements of these dynamic networks.

g. Physical Vulnerability

Another challenge in ad hoc networks is the physical vulnerability of nodes. In a mobile ad hoc network nodes are mobile and often small devices. This contributes to a higher probability of being capture or compromised when compared to traditional wired networks with stationary entities [Lidong &Zygmunt, 1999]. This means that wireless ad hoc networks are more prone to insider attacks and security mechanisms and must be designed with this in mind. An inside attacker could analyze the node to gain secret keying information or use the node to compromise other nodes. The same threats exist in wired formal networks. Although they may rely on a secure host to detect and recover compromised nodes. Sensitive security information may also be stored on that host, minimizing the consequences upon the network if a single node is captured. In an attempt to enhance security within hybrid ad hoc networks [Salem et al, 2005] a fixed architecture is combined with a volatile distributive architecture.

2.3 Security objectives and services

Securing mobile ad hoc networks requires certain services to be met. A security service is a made available by a protocol which ensures sufficient security for the system or the data transferred. The security objectives for mobile ad hoc networks are similar to that of fixed wired networks. The security objects are described in six categories, adapted from discussions in [Stalling, 2003]:

- Authentication
- Access Control
- Data Confidentiality
- Data Integrity
- Non-repudiation
- Availability Services

2.4 Attacks

Threats or attacks upon the network come from entities. They are known as adversaries. Mobile ad hoc networks inherit all the threats of wired and wireless networks. With these networks' unique characteristics, new security threats are also introduced [Zhou & Haas, 1999]. Before the development of security protocols, it is essential to study the attacks associated with these unique networks.

a. Attack characteristics

Attacks will be launched against either the vulnerable characteristics of a mobile ad hoc network or against its security mechanisms. Attacks against the security mechanism in all types of networks, including mobile ad hoc networks, include authentication and secret key sabotage. Mobile ad hoc networks have distinctive characteristics, as identified in Section 2.2. Attackers are expected to target these points of vulnerability, for example the multi-hop nature of communication routes. The attacks are classified by their different characters. The attacks, accordingly, are classified as follows:

Passive and Active Attacks

Security attacks can be classified by the terms active and passive [Stalling, 2002]. Passive attacks attempt to steal information from the network without altering the system resources. Examples of passive attacks include, eavesdropping attacks and traffic analysis attacks. It is difficult to detect passive attacks as they leave no traceable affect upon the system resources or network functionality. Although the results or the need for securing against these attacks may not be monitored or visibly present, it is still a priority to protect networks from these seemingly harmless attacks, particularly in a military context. Concerning this point, Bruce [Bruce, 2003] mentioned: "*If security is too successful, or perfect then the security expenditures are seen as wasteful because success is too invisible*". However, Schneier assures one that, despite the lack of visible results, the need to secure information still exists.

Active attacks attempt to modify system resources or network functionality. Examples of these attacks are message modification, message replay, impersonation and denial of service attacks.

Insider and Outsider Attacks

Malicious nodes are not authorized participants in the network, which launch outsider attacks. Impersonation, packet insertion, and denial of service are some examples of outsider attacks. In contrast to outsider attackers, inside attackers are more difficult to defend against. Inside attacks are launched from nodes which are authorized participants in the network. Insider attacks are common in pure mobile ad hoc network, where any user can freely join or exit. Security mechanism become vulnerable when participates are malicious and the confidentiality of keying information can be compromised. Thus, an advantage of the non-repudiation and authentication techniques, malicious insider nodes can be identified and excluded.

Layer Attacks

There are threats at each layer of the mobile ad hoc network communication protocol. The physical layer is vulnerable to passive and active attacks. The attacks found at the physical layer are as follows: eavesdropping; denial of service; and physical hardware alterations. Encrypting the communication links and using tamper-resistant hardware helps to protect the physical layer. However, at the data link layer adversaries can flood the communication links with unnecessary data to deplete network resources. Security mechanisms that provide authentication and non-repudiation can prevent this, as they allow invalid packets transfers to be identified. At the application layer messages are exchanged in an end-to-end manner using wireless multi-hop routes established by the network layer. The wireless multi-hop routes are invisible to the application layer. Conventional security techniques used for wired networks can be used to prevent expected attacks upon the application layer. The application layer is dependent upon the network layer to provide secure routes between the two communicating parties.

The network layer provides a critical service to the mobile ad hoc network, and the routing protocol. In the context of trust and security, the provision of secure routes is one of the most vital elements for trust establishment.

b. Attack Types

The different types of attacks are identified and described below. While there is a focus on the networking layer, attacks such as impersonation and denial of services can occur on any layer.

Wormhole attack

In a wormhole attack a compromised node receives packets at one place in the network. The attacker tunnels the packets to another destination (i.e. an external attacker) in the network, where the packets are resent back into the network [Qian & Li, 2007]. The tunnel created by the adversary is known as a wormhole. A wormhole allows adversaries to disturb the routing protocol, by intercepting routing messages and creating denial of service attacks. If the routing mechanism is not protected against such an attack mobile ad hoc routing protocols may fail to find valid routes.

Black hole attack

During route discovery a malicious node may falsely advertise itself as possessing the optimal route to the requested destination. The adversary, therefore, attracts all routing messages. The attacker then creates a black hole attack by dropping all routing packets, and disrupting the routing protocol and discovery phase.

Byzantine attack

During this type of attack a malicious node, or a group of malicious nodes, will launch attacks on the routing protocol. The aim is to direct routing packets to follow: non-optimal routes; routing loops; and selective dropping of packets [Awerbuch et al, 2002]. Byzantine behaviour is difficult to detect. A network could be operating with byzantine failures and be unaware of the attack on its routing mechanism.

Eavesdropping

An eavesdropping attack involves message or routing packet monitoring. It is a passive attack on the mobile ad hoc network. Eavesdropping attacks are performed by adversaries and can reveal confidential information about the network regarding: its topology; geographical locations; or optimal routes in the network. Attackers can use this information to launch other attacks at identified points of vulnerability. All networks are prone to passive eavesdropping attacks. It is the nature of wireless, mobile ad hoc networks that make them more vulnerable. In wireless networks adversaries do not need a physical wired communication link to monitor the routing packets. The wireless communication medium allows for any users, within range, to analyze the traffic. Attackers can also exploit the multi-hop nature of routes in mobile ad hoc networks. An adversary can position itself along a route path and forwarding the routing messages along the multi-hop path. This allows adversaries to also analyze every packet that is forwarded along the path. Eavesdropping is a common problem in networks and encryption techniques can protect routing protocols from these attacks.

Packet Replay

Like eavesdropping, replay is a passive attack where data is captured by monitoring adversaries. Old routing messages are then retransmitted to other nodes disturbing the routing process. Adversaries can, therefore, cause other node’s routing tables to be updated with outdated information. Malicious attackers can also record authorized routing messages and replay them to gain unauthorized access to protected nodes.

Resource consumption attack

Mobile ad hoc nodes are restricted by their limited resources. Attackers exploit this by launching attacks that consume a node's resources hindering them from network participation. Resources targeted by attackers are: bandwidth, computational power and battery life.

Sleep deprivation attacks, are resource attacks which are, specifically aimed against mobile ad hoc node's battery power. Node's attempt to save power by going into a sleep mode, where a periodic scanning occurs and less battery power is used. Sleep deprivation attacks prevent nodes from going into sleep mode therefore draining the battery life and disabling the node itself. Attackers will flood a target node with redundant routing requests or routing packets to be processed, thereby keeping the node and its resources unnecessarily busy.

Packet replication is another type of resource attack where adversaries duplicate out of date packets and re-transmit them. This not only consumes battery life, bandwidth and computational power, but also disrupts the routing protocol.

Sleep deprivation attacks, flooding attacks and packet replication result in the depletion of precious resources. If this is not protected against, it will result in nodes and services becoming unavailable in the network.

Routing Table Poisoning

Malicious nodes will target the routing table in an attempt to sabotage the establishment of routes. One such attack is the routing table poisoning attack where malicious nodes send counterfeit routing updates or modify existing routing updates. This results in conflicting link information, unnecessary traffic congestion or denial of service.

Rushing attack

Mobile ad hoc networks that use on-demand routing protocols are vulnerable to rushing attacks [Hu et al, 2003a]. On-demand routing protocols, such as AODV [Perkins et al, 2003] and DSDV [Perkins & Bhagwat, 1994], use route request messages to discover the optimal route to a destination node. The network is flooded with route request messages. These messages are forwarded until the optimal route is found between the source and destination nodes. An adversary that receives a route request performs a rush attack by hurriedly flooding the network with that route request before other nodes, receiving the same route request, can respond. When other nodes receive the legitimate routing request, it is assumed to be a duplicate of the request which is distributed by the adversary, and the legitimate routing request is dropped. Therefore, the adversary will become part of the route that is discovered. This will result in an overall, insecure route.

Selfish attack

Misbehaving nodes will act in a greedy or selfish manner, resisting cooperating or participating in the network operations. This is a denial of service and the attack causes the nodes to refuse to make their resources available. Selfish nodes do not cooperate in network operations that do not benefit them. Rather they conserve their limited resources, such as battery life. Nodes may refuse to forward route request packets or turn off their devices when they are not transmitting data. The distributive architecture and multi-hop nature of mobile ad hoc networks means the network relies upon node cooperation [Molva & Michardi, 2003]. A security protocol should ensure fair distribution of network operation in order to provide reliable network services, and prevent node's resources becoming depleted because of selfish node attacks.

Impersonation

Impersonation attacks are also known as masquerading or spoofing attacks. The attacks occur when adversaries take the identity of an authorized node and breach the security of the network. Masquerading nodes are able to receive routing packets destined for other nodes. Mobile ad hoc networks can help protect against impersonation attacks by authenticating their routing messages.

Pure mobile ad hoc networks are more vulnerable as they have no access control. If there is no strong binding between the physical entity and the network identity, malicious nodes can adopt different identities. A severe attack which is prone to mobile ad hoc networks is the Sybil attack [Hashmi & Brooke, 2008] [Douceur, 2002]. A single adversary node launches a Sybil attack by adopting multiple identities and participating in the network with all identities at once. The result of such an attack gives the attacker a majority vote or considerable control in the network.

2.5 Security model

A security model for mobile ad hoc networks is illustrated, in general terms, in Figure 2. A message *M* is to be transmitted from the source *A*, across a network of nodes, to a destination node *B*. The two entities who are primary participants must collaborate for the transaction to occur. A routing protocol establishes a multi hop route between the primary

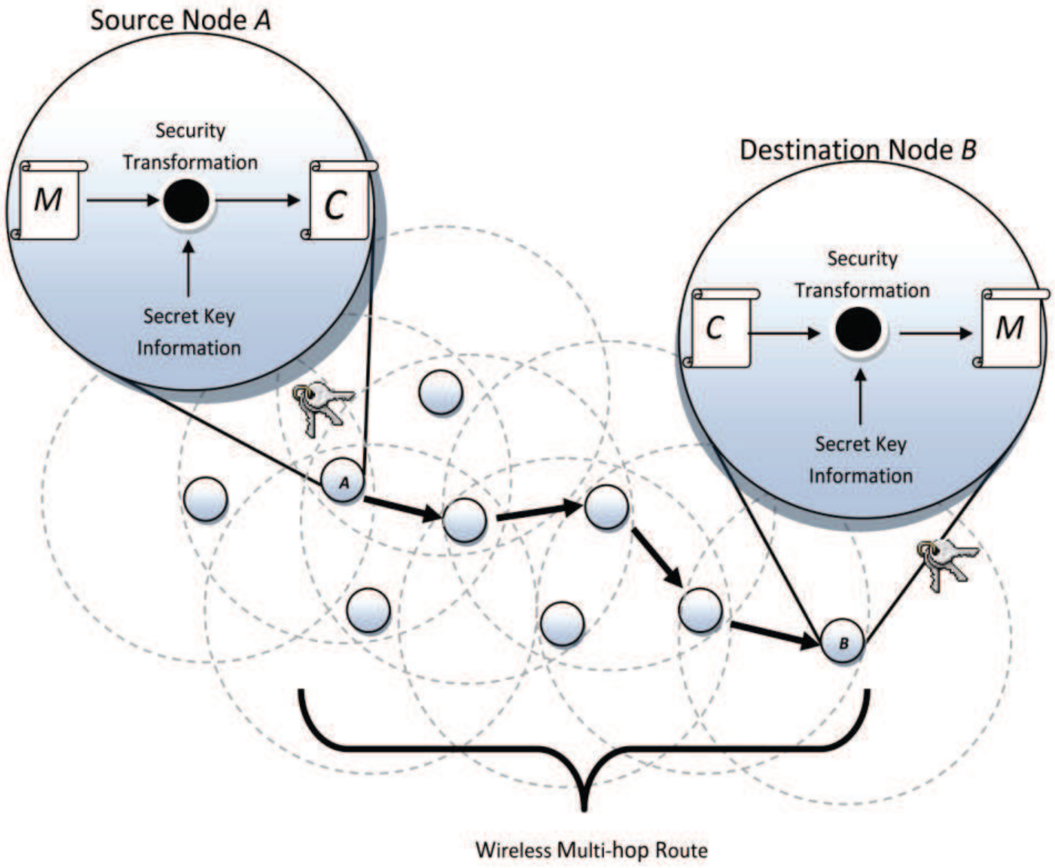


Fig. 2. General Security Model

participants. The multi-hop route will involve secondary participating nodes. Security is provided by two accompanying techniques: a security related transformation applied to the message (resulting in an encrypted message *C)* and secret keying information shared by the principal participants.

The general mobile ad hoc security model shows four basic tasks for a security mechanism:

1. The design of a security algorithm.
2. Generation of secret keying material used in conjunction with this security algorithm.
3. Distribution of secret keying material.
4. Protocol for the participants to follow which will achieve the required security services.

Tasks 1 and 2 deal with the cryptographic algorithm used to provide security services. It is widely recognized that existing cryptographic technology can provide sufficiently strong security mechanisms to ensure routing message confidentiality, authentication and integrity. The establishment of these security mechanisms is a dynamic problem in wireless ad hoc networks, as this network cannot adopt the same approaches of its wired predecessors. The focus of this chapter is upon tasks 3 and 4: the establishment of the security protocols in the mobile ad hoc environment.

3. Key management in mobile ad hoc networks

In Section-2 we discussed the different types of attacks upon wireless ad hoc networks. In this section the techniques used to prevent these malicious attacks and specifically key management techniques, will be looked at. Security solutions which use cryptographic techniques rely on proper key management to establish trust. This chapter focuses upon key management which aids these cryptographic solutions.

3.1 Description of key management

In any communication network, the cryptographic network security is dependent on proper key management. Mobile ad hoc networks vary significantly from standard wired networks. A specific efficient key management system is required to realize security in these networks. Key management is defined as a set of procedures employed to administrate the establishment and maintenance of secure key base relationship. The purposes of key management, as stated by Menezes et al [Menezes et al, 1996b], is to:

1. Initialize system users within a network.
2. Generate, distribute and install keying material.
3. Control the use of keying material within the network.
4. Update, revoke, destroy and maintain keying material.
5. Store, backup and recover keying material.

Key management systems are responsible for the secure distribution of keys to their intended destinations. Keys which are required to remain secret must be distributed in a way that ensures confidentiality, authenticity and integrity. For example, in symmetric key cryptography both, or all, the participants must receive the key securely. For asymmetric key cryptography, the key management system must ensure that private keys are kept secret and only delivered to the required, authorized participants. Public keys do not require confidentiality but, authentication and integrity is vital. The key management system must protect confidentiality and authenticity of the keys. This system must also prevent unauthorized use of keys, for example the use of keys which are out-dated and invalid.

Cryptographic algorithms can provide confidentiality, authentication and integrity. However, the primary goal of key management is to guarantee that the secret keying material is shared among the specific communicating participants securely. There are several methodologies of sharing the keying material. The main approaches are: key transport; key arbitration; key pre-distribution; and key agreement [Menezes et al, 1996b].

a. Key Transport

In a key transport system, one entity generates keys, or obtains keying material, and securely transports them to other entities in the network. The simplest key transport method is the key encrypting key method (KEK). This method assumes a prior shared key exists among the participating nodes. The prior shared key is used to encrypt new keys and transport them to all participating nodes. Prior shared keying relationships cannot be assumed in networks, especially in mobile ad hoc networks. If a public key infrastructure exists, then the new keys can be encrypted by the respective receiver's public key and transported without the existence of prior keying relationships. This approach assumes the existence of a trust third party (TTP) member which transports all the keying material. In pure mobile ad hoc networks a TTP member would not be available. Shamir's three-pass protocol [Shamir, 1979] is a key transport method, without prior shared keys.

b. Key Arbitration

A key arbitration system is a division of key transportation. In key arbitration a central arbitrator is assigned to create and distribute keys to all participants. The arbiter is often a wired node with no resource constraints. In mobile ad hoc networks nodes are wireless with resource constraints. The arbiter would be required to be online throughout the network communication and be accessible to every member in the network. This is difficult in mobile ad hoc networks because of the resource constraints such as: bandwidth; transmission range; and energy. A solution to these potential problems is a distributive system, where the arbiter is replicated at different nodes. Simple replication of the arbiter has severe resource expenses on certain nodes and creates multiple points of vulnerability in the network. If a single replicated arbiter is compromised the entire network can be at risk.

c. Key Pre-distribution

Keys are distributed to all participating member before the start of communication. Key pre-distribution requires prior knowledge of all participating nodes. Its implementation is simple and involves much less computation than other schemes. This method is suitable for mobile ad hoc sensor networks, as they have highly restrictive resource capabilities. The set of sensor nodes is also established before the network is deployed and data is tracked. Once the network is deployed there is no service which allows for new members to join or for keys to be changed. This method is extended by allowing sub-groups of communication to form in the network. Similarly, the decision is made prior to deployment, and not during communication.

d. Key Agreement

Key agreement is used to enable two participants to agree upon a secret key. In this way, keys are shared and establish a secure communication line over which a session can be run. Key agreement schemes are often based on asymmetric key cryptography and have high computational complexity, but little pre-configuration required. The most widely used key agreement scheme is the Diffie-Hellman key exchange [Steiner et al, 1996]. This is an asymmetric keying approach based on discrete logarithms.

3.2 Key management in mobile ad hoc networks

Ad hoc wireless networks have unique characteristics and challenges, which do not allow the simple replication of conventional key management methods that are used for wired networks. Mobile ad hoc network's lack of infrastructure poses the greatest threat to the establishment of a secure key management scheme. Fixed infrastructure such as: a trusted third party member; an administrative support or certificate authority; dedicated routers; or fixed reliable communication links, cannot be assumed in wireless ad hoc networks. Unique solutions are required for such unique networks. The focus of this chapter is around the investigation of the existing key management schemes for mobile ad hoc networks.

Key management schemes are investigated with regard to: functionality; scalability; availability; security services; efficiency; and computational cost. A key management solution, which is scalable, will effectively provide security services in a network which dynamically changes in size, as nodes join and leave the network. Availability is essential for a network whose topology is rapidly changing. Nodes should have easy access to authority members and keying services. A high priority is given to a key management solution that can successfully and efficiently provide crucial security services for the keying material. Such services include: key confidentiality; key authenticity; key integrity; and fresh key updates. These services are congruent with the security services described in Section 2.

Of the existing key management solutions, asymmetric cryptography is predominately used when managing trust via a public key infrastructure (PKI) of some sort. Existing PKI schemes utilize either the: hierarchical or web-of-trust model.

a. Hierarchical Trust models

The hierarchical trust models are more structured, as they use a PKI and a certificate authority as a source of trust. The certificate authority (CA) is a trusted entity used to verify; issue; and revoke certificates, therefore enabling successful public key cryptography. A key management service for public key cryptography would include the certificate authority service which has a public key, *K*, and private key, *k*. The CA's public key is distributed to all the nodes in the network. The nodes know that any certificate signed by the CA's private key may be trusted. Each node also has its own public/private key pair, which allows for nodal communication. The CA stores the public keys of all the network nodes and distributes the respective keys to the nodes that request to setup a secure communication with another node [William, 1999]. A fixed CA is not considered in this investigation, due to the limitations caused by no TTP.

The CA distributes trust in a hierarchical manner, as seen in Figure 3. A root CA issues certificates to delegated CA's or end users. The CA can issue certificates to user nodes or other CA nodes. The PKI X.509 framework is an example of such an infrastructure [Stalling, 2003].

The following types of hierarchal trust models have been investigated in the context of mobile ad hoc networks:

1. *Off-line trusted third party models:* use a trusted outside entity to achieve a large portion of the key management tasks.
2. *Partially distributed certificate authority models:* distribute the functionality of the CA to a small set of nodes.
3. *Fully distributed certificate authority models:* are self organized models, which are similar to the previous distribute to the CA. However, this model is across the entire network in a self organized manner.

4. *Cluster based model:* is a special kind of hierarchical trust in the form of group authentication, where *clustered groups* of nodes are treated as single trust entities and authenticated as a group.

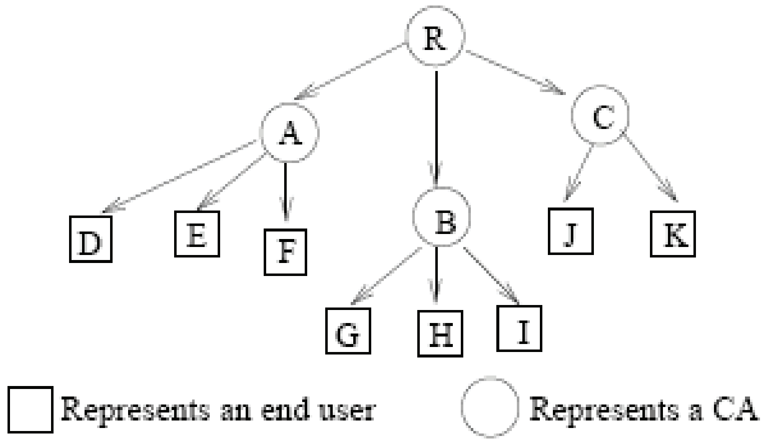


Fig. 3. Hierarchical trust

b. Web-of-Trust Models

The Pretty Good Privacy model (PGP) [Abdul-Rahman, 1997], also known as a "web-of-trust-model", enables nodes to act as independent certification authorities. There is no distinction between a CA and an end user node. Nodes provide individual trust opinions of other nodes, thereby creating a "web of trust", as illustrated in Figure 4. Each user node is the "centre of its own world" and is responsible for certificate management. The advantage of a PGP model is its dynamic, autonomous nature, which is seemingly ideal for application in decentralized environments such as ad hoc networks [Davis, 2004].

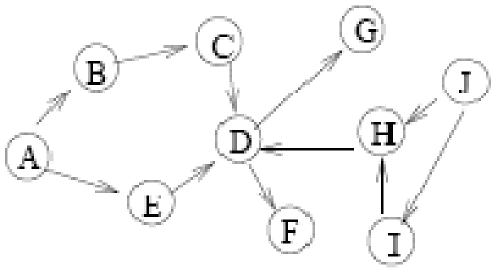


Fig. 4. PGP web-of-trust

Certificates are issued by the nodes themselves. However, a public certificate directory is required for their distribution. This directory is often located at an online, centralized, trusted third party entity. This makes the PGP model unsuitable for ad hoc network application. Steps are needed to be taken to localize such directories and realize certificate distribution. The autonomous nature of the "web-of-trust" model means that it is more susceptible to malicious attackers than to more structured networks. For example, if one entity is compromised a corrupt set of certificates is filtered throughout the network. The self issued certificate model is investigated as a foundation for PGP based solutions in mobile ad hoc networks. Figure 5 illustrates the key management solutions investigated in this chapter.

3.3 Off-line trusted third party models

A progress trust negotiation scheme was introduced by Verma [Verma et al, 2001]. It is a hierarchical trust model where authentication is preformed locally, but an off-line trusted third party performs trust management tasks like the issuing of certificates. The off-line trusted third party also manages the certificate revocation process. This scheme is extended through a localized trust management scheme proposed by Davis [Davis, 2004]. Davis attempts to localize Verma's solution. The only trust management task that is not implemented locally is the issuing of the certificates.

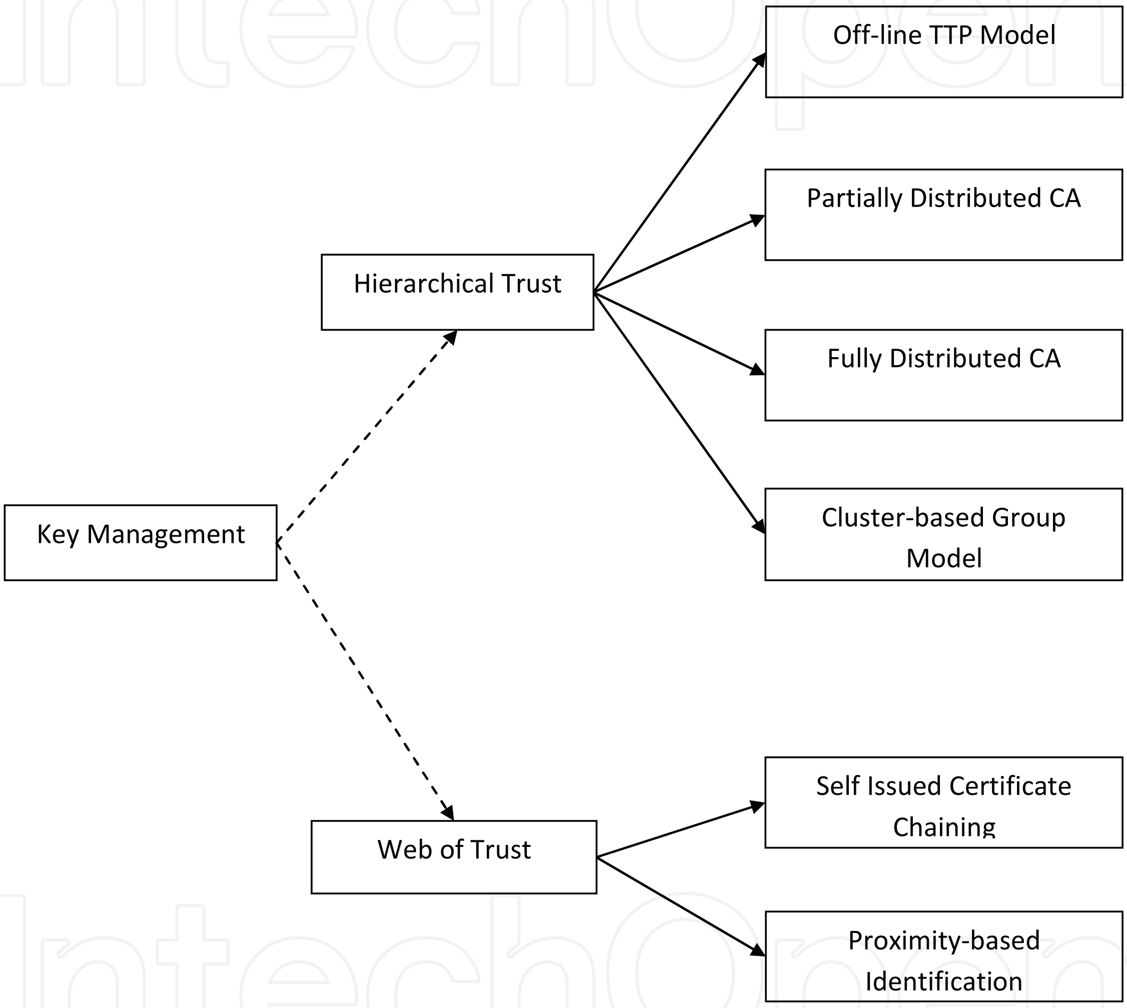


Fig. 5. Key Management Solutions

a. System Overview

Each node possesses its own private key and the trusted third party's public key. The maintenance of these keys is the responsibility of each node. Trust is established when the trustor provides the trustee with a certificate that has not expired, or has not been revoked and the trustee can verify it with the trusted third party's public key (possessed by the trustee). Furthermore, to realize certificate revocation, each node must possess two certificate tables: a status and profile table. The profile table, illustrated in Figure 6, describes the conduct or behaviour of each node. The status table describes the status of the certificate, i.e. revoked or valid. These two tables are maintained locally by the nodes themselves, with the purpose of maintaining consistent profiles.

Davis's scheme is a fully distributed scheme. It requires that a node broadcasts its certificates and its profile table to all the nodes in the network. It also requires that each node's profile table be kept updated, and distributed with synchronization of data content. The profile table contains information from which the user node may define if a certificate can be trusted or of it must be revoked. Node *i*'s profile table stores three pieces of data:

1. *Accusation info:* the identity of nodes that have accused node *i* of misbehaving.
2. *Peer n ID:* the identity of nodes that node *i* has accused, acting almost as a CRL (certificate revocation list).
3. *Certificate status:* a 1-bit flag indicating the revocation status of the certificate.

The fully distributed information in the profile tables should be consistent. If there is any inconsistency detected, an accusation is expected to be launched against the node in question. Inconsistent data can be defined as data which differs from the majority of data.

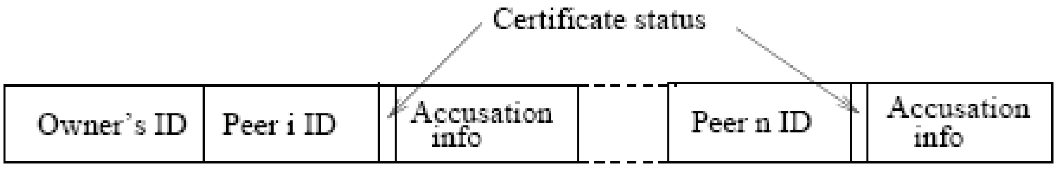


Fig. 6. Profile Table

The status table is then used to calculate the certificates status, i.e. revoked or not revoked. The node *i*'s status table stores and analysis the following factors: A_i (total number of accusations against node *i*); a_i (total number of accusations made by node *i*) ; *N* (expected maximum number of nodes in the network). These factors are used to calculate the weight of node *i*'s accusation and the weight of other nodes accusations against node *i*. A revocation quotient is then calculated, R_j, as a function of the sum of the weighted accusations. It is then compared to a network defined revocation threshold R_T. If $R_j > R_T$ then the node *i*'s certificate is revoked.

b. Analysis

This scheme uses a hierarchical trust model which relies upon an off-line trusted third party for aspects of key management. The off-line trust third party is to be resident as a trusted source if required. This scheme assumes the existence of a trusted off-line entity which initializes certificates, and securely distributes them amongst the network participants. This scheme is a pre-distributive key exchange model. It provides robust security; however, its implementation is more realistic within a hybrid infrastructure. A key management scheme with a hybrid infrastructure is a scheme which makes use of both wired and wireless architecture. A wired trusted off-line node performs all or a portion of the key management services to maximise security and efficiency. Hybrid infrastructures allow for greater security and a simple solution to the central problem of key distribution in mobile ad hoc networks.

Verma and Davis's solution does not specify that a wired node be the off-line authority for key pre-distribution. Nevertheless, a separate trusted entity capable of intense computation, high security and network distribution must exist for the success of Verma and Davis's model. Such assumptions cannot be made in pure mobile ad hoc networks. The hybrid nature of Davis's solution is displayed in Figure 7.

Verma localizes the task of authentication. Davis goes one step further by localizing the revocation module of the scheme by proactively maintaining accusation information in profile tables and locally, calculating revocation decisions. This scheme mitigates against malicious accusation exploits. This could result in a node being revoked based on single malicious offender's broadcast information. To solve this problem one must not treat all accusations equally, but rather use a sum of weighted accusations, which are calculated before the node is revoked. Davis's scheme succeeds in taking steps toward self-organization in ad hoc network trust establishment as it provides a protocol that enables revocation of certificates, without continual trusted third party involvement.

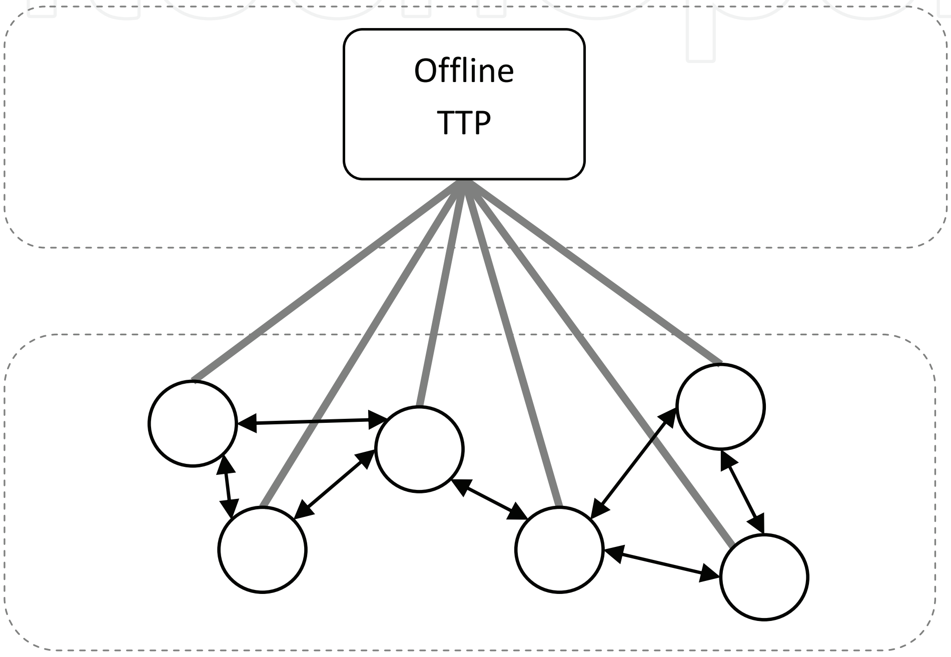


Fig. 7. Hybrid progressive trust negotiation scheme

3.4 Partially distributed certificate authority

The solution proposed by Zhou and Haas [Zhou & Hass, 1999] allows for the functionality of the certificate authority to be shared amongst a set of nodes in the network. This solution aims to create the illusion of an existing trusted third party. Zhou and Haas's proposal in 1999 was instrumental in the initial research of key management solutions for ad hoc networks. This approach has been extended to incorporate the heterogeneous nature of nodes in [Yi & Kravets, 2001].

a. System overview

The CA's public key, *K,* is known by all nodes (*m*) and the CA's private key, *k,* is divided and shared by *n* nodes where *n* < *m*. The distributed CA signs certificates by recreating the private key via a *t* threshold group signature method. Each CA node has a partial signature. The CA's signature is successfully created when *t* correct partial signatures are combined, at a combiner node. To prevent the distributed CA nodes from becoming compromised and the authentication becoming compromised, a preventive proactive scheme is implemented as to refresh the CA nodes. A simple partially distributed CA system is illustrated in Figure 8.

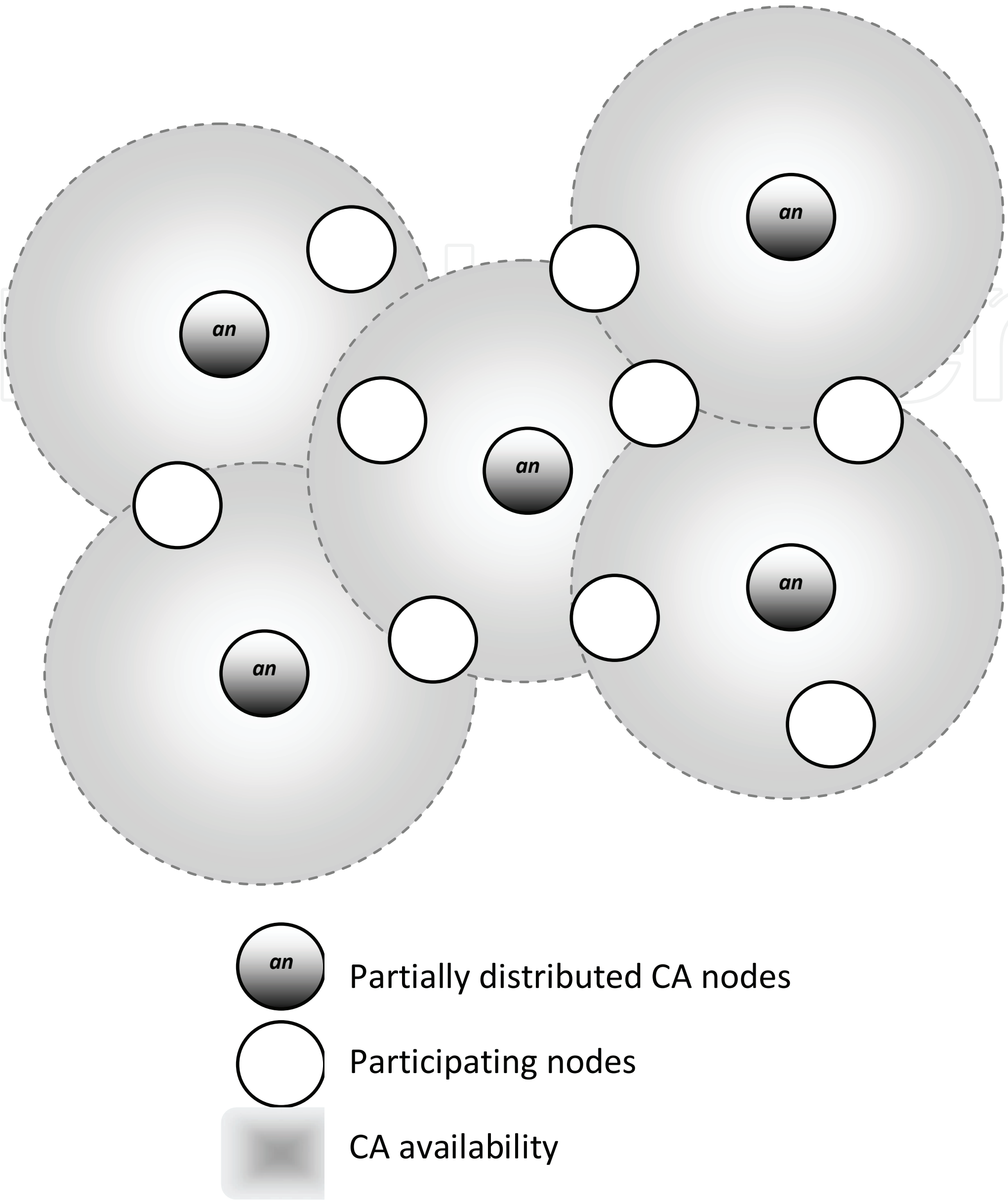


Fig. 8. Partially Distributed Certificate Authority

b. Threshold Scheme

Threshold cryptography is used to share the CA service between nodes. A threshold cryptography scheme allows the sharing of cryptographic functionality. A *(t-out-of-n)* threshold scheme allows *n* nodes to share the cryptographic capability. However, it requires *t* nodes, from the *n* node set, to successfully perform the CA's functionality jointly. Potential attackers need to corrupt *t* authority nodes, before being able to exploit the CA's functionality and analyze secret keying information. Therefore, a *(t-out-of-n)* threshold scheme tolerates *t-1* compromised nodes, from the *n* node set [Aram et al, 2003].

When applying threshold cryptography to the shared CA problem, the CA service is shared by *n* nodes across the network called authority nodes. The private key *k*, crucial for digital signatures, is split into *n* parts ($k_1,k_2,k_3,\ldots,k_n$) assigning each part to an authority node (*an*). Each authority node has its own public key, $K_{n,}$ and private key, $k_{n,}$ (as seen in Figure 9).It stores the public keys of all the network nodes (including other authority nodes). Nodes wanting to set-up secure communication with node *i* need only request the public key of node *i* (K_i) from the closest authority node - therefore increasing the CA's availability. For the CA service to sign and verify a certificate, each authority node produces a partial digital signature using its respective private key, $k_{p,}$ and then submit the partial digital signature to a combining node. Any node may act as a combiner in the ad hoc network. The partial digital signatures are combined at a combiner (*c*) to create the signature for the certificate, *t* correct partial digital signatures are required to create a successful signature. Therefore, protecting the network against corrupt authority nodes, up to *t-1* corrupt authority nodes may be tolerated [Lidong & Zygmunt, 1999].

For example, Figure 10 shows a (*2-out-of-3*) threshold scheme where the message *m* is signed by the CA, two partial signatures (*PS*) are accepted, while the third (an_2) was corrupted. The partial signatures meet the threshold requirements and the partial signatures are combined at *c* and applied to the message.

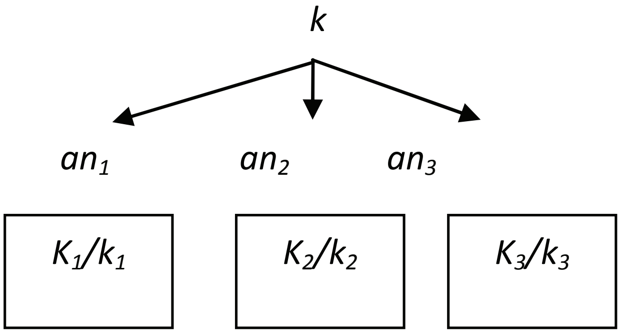


Fig. 9. (2-out-of-3) Threshold Key Management

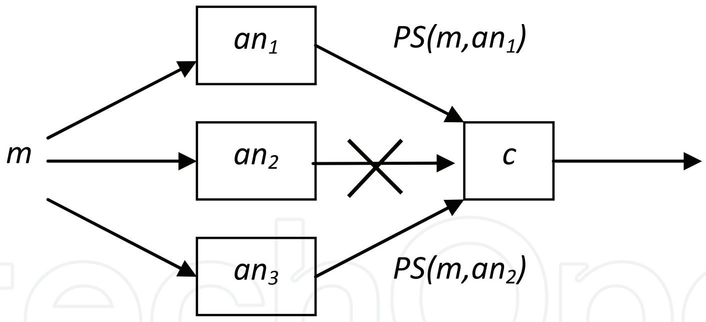


Fig. 10. (2-out-of-3) Threshold Signature

c. Proactive security

Threshold cryptography increases the availability and security of the network by de-centralizing the CA. Security is maintained with the assumption that all CA authority nodes cannot be simultaneously corrupt.

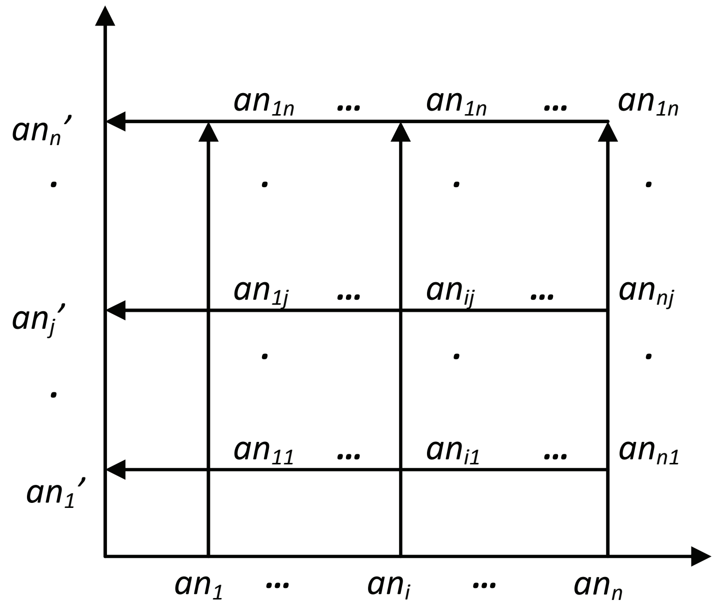
It is possible for a malicious attacker to compromise all the CA's authority nodes over time. An adversary of this type is then able to gain the CA's sensitive keying information. Proactive schemes [Van der Merwe & Dawoud, 2004] [Herzberg et al, 1997] [Frankel et al, 1997] [Jarecki, 1995] are implemented to avoid such adversaries.

A proactive threshold cryptography scheme uses share refreshing. This enables CA authority nodes to compute new key shares from old ones, without disclosing the CA's public/private key. The new key shares make a new *(t-out-of-n)* sharing of the CA's public/private key pair. These are independent of the old pair [Herzberg et al, 1995].

Share refreshing relies on the following mathematical property:

If (s_{11} , s_{21} , … ,s_{n1}) is a *(t-out-of-n)* sharing of k_1 and (s_{12} , s_{22} , … ,s_{n2}) is a *(t-out-of-n)* sharing of k_2, then ($s_{11} + s_{12}$, $s_{21} + s_{22}$, … ,$s_{n1} + s_{n2}$) is a *(t-out-of-n)* sharing of $k_1 + k_2$. Therefore if k_2 is 0, then we get a new *(t-out-of-n)* sharing of k_1.

The share refreshing scheme is applied to a threshold CA. A threshold CA is a *(t-out-of-n)* system that shares the CA's private key *k* among *n* authority nodes (an_1 , … , an_n) each with a share of the CA's private key. To generate a new *(t-out-of-n)* sharing (an_1', … , an_n') of *k,* each authority node an_i generates sub-shares (an_{i1} , an_{i2} , … , an_{in}) a *(t-out-of-n)* sharing of 0, which represents the *i*'th column, as seen in Figure 11. Each sub-share an_{ij} is sent to the authority node an_j. When authority node an_j has received all sub-shares (an_{1j} , an_{2j} , … , an_{nj}), which represents the *j*th row, seen in Figure 11, it then generates its new share an_1' by using the mathematical property described above.



an_n'
an_j'
an_1'
an_{1n} … an_{1n} … an_{1n}
an_{1j} … an_{ij} … an_{nj}
an_{11} … an_{i1} … an_{n1}
an_1 … an_i … an_n

Fig. 11. (t-out-of-n) Share Refreshing

The communication of the sub-shares requires a secret redistribution protocol [Desmendt & Jajodia, 1997] [Chor et al, 1985] to ensure secure transmission. Note that share refreshing does not change the CA's private key pair. Share refreshing may occur periodically and be extended to occur upon events. These events can include the detection of compromised nodes or a change in network topology. Therefore, the key management service is able to transparently adapt itself to changes in the network and maintain secure communication.

d. Heterogeneous Extension

An extension to Zhou and Haas's scheme can be seen in the Mobile Certificate Authority (MOCA) scheme by Yi and Kravets [Yi & Kravets, 2003]. The MOCA scheme also uses threshold cryptography to implement a public key, which is a partially distributed certificate authority solution. The functionality of the certificate authority is distributed to *n* nodes, called MOCAs. The assumption is made that all nodes have heterogeneous visible qualities. These visible qualities act as initial trust evidence and are used when selecting the

MOCA nodes to distribute authority. Such visible evidence can include: computational power; physical security; or position. This evidence is based on a trust decision and authority distributed, accordingly. Similar to Zhou and Haas's scheme, nodes require *t+1* partial signatures from a set of *n* MOCAs to allow for certificate verification and trust relationship establishment, with a threshold of *t*. The MOCA scheme further builds on Zhou and Haas's solution by adding a revocation of certificates. Certificate revocation lists are stored at each MOCA. For certificates to be revoked, *t+1* MOCAs must sign a revocation certificate request with *t+1* partial signatures from the MOCAs. Once the partial signatures are gathered, the certificate revocation list is updated. Malicious nodes wanting to unnecessarily revoke another node's certificate can only do so with the approval of *t+1* trusted MOCAs, therefore ensuring the reputation of each node's certificate.

e. Analysis

This solution demonstrates some of the problems of an ad hoc network. Despite its obvious weaknesses, it is noted as one of the earliest key management solutions to ad hoc networks.

The partial distributive scheme proposed by Zhou and Haas requires that an off-line TTP member exists at the initialization phase in order to establish the distributive CA. The off-line TTP: generates the threshold private key; shares it among the appointed CA authority nodes; and distributes the CA's public key to all participating nodes in the network. All certificate related tasks including signatures, generation, distribution, refreshing and revocation, are performed by the participating nodes without the involvement of a TTP. The off-line TTP is not as involved in Verma [Verma et al, 2001] and Davis's [Davis, 2004] proposals. However, in spontaneous ad hoc networks such a trusted entity cannot be assumed at initialization.

The advantage of distributing the CA allows for the functionality of the CA to be distributed among the nodes. This avoids single point attacks and allows the computational overhead of the CA's services to be distributed. Although the CA is distributed, it still remains centralised between a few nodes.

The centralization of authority creates availability issues. The availability issues are sensitive as communicating nodes require communicating with *t* authority nodes before acquiring a signature. The CA's availability is dependent on the threshold parameters *t* and *n*. These parameters must be selected to provide a suitable trade-off between: availability; security; and cost of computation. The larger the threshold (*t*), the higher the security, but, the availability will pay the cost. The centralization of authority also results in a select group of nodes carrying the burden of security computations. This breaks the value of fair distribution in a network.

This solution requires that the CA authority nodes store all the certificates issued, which necessitates a costly synchronization mechanism. Furthermore, a share refreshing or proactive method is required. This is achieved by using a secret redistribution protocol [Desmendt & Jajodia, 1997]. With this in place, it is, therefore, certain that all the CA authority nodes are not compromised. The procedure of synchronization, updating and proactive refreshing is costly to resource constrained nodes.

Another potential problem is related to network participants addressing the CA authority nodes. A node requesting a service from the CA entity is required to contact *t* out of *n* nodes. The CA can then be given a multicast address and participating nodes can multicast their requests to the CA. The CA authority nodes can then unicast replies to the requesting participant. In ad hoc networks, which do not support multicasting, a participating node can broadcast its request. This approach is more common in mobile ad hoc networks, despite its potential of a large amount of network traffic.

Zhou and Haas's partially distributed certificate authority approach provides much of the groundwork for future solutions through the implementation of threshold cryptography in ad hoc networks.

3.5 Fully distributed certificate authority

The threshold scheme, investigated in [Luo & Lu, 2000] [Luo et al, 2002], uses ideas proposed by the partial distributive threshold scheme, found in [Lidong &Zygmunt, 1999]. *Luo* and *Lu* propose a scheme which embraces the distribution of the CA. In a network of *m* nodes, the network and security services are shared across *m* nodes. Therefore, a fully distributed system is realized, as seen in Figure 12. This scheme further differs from [Lidong &Zygmunt, 1999] in that there is no need to select specialized nodal authorities, as all nodes perform this role. Like the partial distributive scheme, the fully distributive scheme includes the use of share refreshing. This allows proactive security against significant nodes that are compromised. This scheme is designed for, and aimed at, long-term ad hoc networks which have the capacity to handle public key cryptography.

a. System overview

The Fully Distributive Certificate Authority scheme is a public key cryptography scheme. It takes the functionality of the certificate authority and distributes it across *m* nodes, where *m* is the total number of nodes in the network. This threshold scheme requires *k* or more nodes to act in collaboration to perform any operations of the CA. The CA's private key is divided and shared among all the participating nodes. This effectively enhances availability and allows nodes that are requesting the CA, to contact any *k* one-hop neighbour nodes. It is assumed that each node will have more than *k* one-hop neighbours [Luo & Lu, 2000]. Therefore, only one-hop certificate communication can occur. This allows for more reliable communication, in comparison with multi-hop communication. It is also easier to detect compromised nodes. Figure 12 illustrates the fully distributive network, where all nodes have a portion of authority in the form of a partial CA signature. Figure 12 shows a network with threshold *k=3,* where nodes *B*, *C* and *D* can find a coalition of partial CA nodes to form a group authentication CA signature. Node *A* is unable to find a sufficient coalition of nodes.

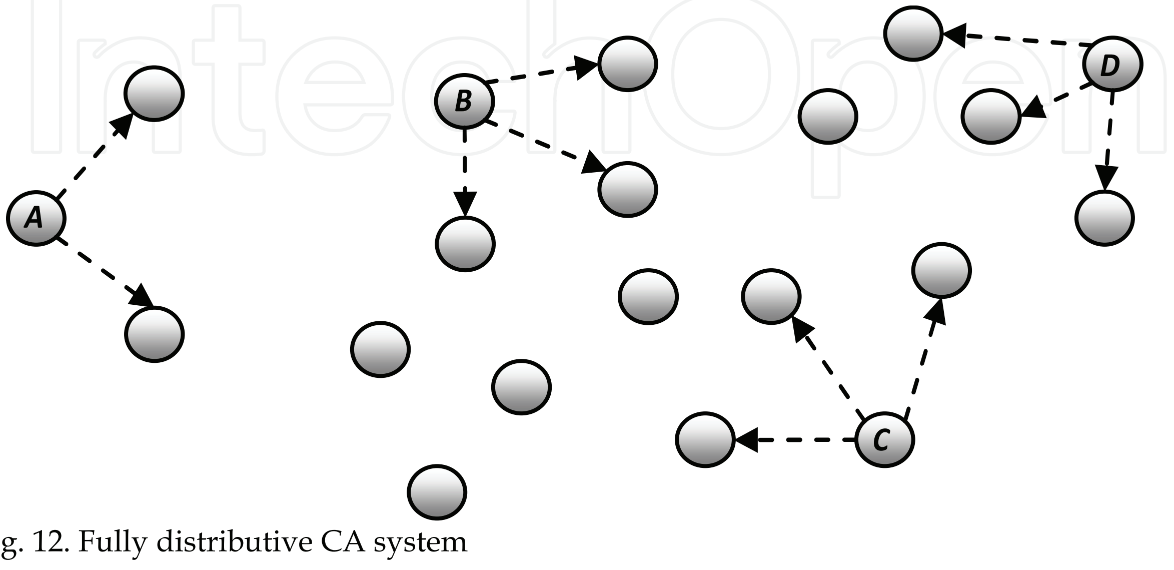


Fig. 12. Fully distributive CA system

b. Off-line Initialization

The initial phase of [Luo & Lu, 2000] [Luo et al, 2002] requires an off-line trusted third party (TTP) to establish the initial set of nodes. The off-line TTP will provide each node *i* with its own: certificate; the CA's public key; and a share of the CA's private key. A certificate is a binding between a nodes ID and its public key. The certificate is signed by CA's private key k_{CA} and can be verified by the CA's public key K_{CA} - which is made available to all the participating nodes. The off-line TTP initialises the threshold private key to the first *k* nodes by the following steps:

1. Generate the sharing polynomial $f(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$ where $a_0 = k_{CA}$
2. Securely distribute node *i* identified by ID_i where $i \in k$ with its secret share $S_i = f(ID_i)$
3. Broadcast *k* public witnesses of the sharing polynomial's coefficients $\{h^{a_0}, \ldots, h^{a_{k-1}}\}$ and then the off-line TTP involvement is over.
4. Each node with ID_i that has received a secret share S_i verifies it by checking the sharing polynomial's coefficients such that $h^{S_i} = h^{a_0} \cdot (h^{a_0})^{ID_i} \cdot (h^{a_1})^{ID_i^2} \cdot \ldots \cdot (h^{a_{k-1}})^{ID_i^{k-1}}$.

After the initial establishment of the shared secret key amongst the first *k* nodes, the TTP is no longer responsible for the full distribution of the CA's private key. The off-line TTP maintains the responsibility of issuing new nodes with their initial certificates binding, and as a result impersonation attacks are prevented.

c. On-line Shared Initialization

New nodes entering the network need to be provided with their own share of the CA private key k_{CA} so that they can be part of the signing process. The participating nodes in the network perform this initialization process, without the interference of an off-line TTP. Shared initialization is modelled on Shamir's threshold secret sharing scheme [Shamir, 1979]. This scheme allows for a culmination of *t* nodes to initialize a joining node, with a share of the CA private key k_{CA}.

A node *i,* already initialized by the off-line authority, can generate a partial secret share $S_{p,i}$ for a joining node *p*. The combination of *k* partial secret shares results in node *p*'s secret share $S_{p.}$ This is a partial share of the CA's private key.

$$S_p = \sum_{i=1}^{k} S_{p,i}$$

Node *i*'s secret share S_i can be derived from each partial secret share $S_{p,}$ which is sent to node *p*. The joining node *p* must not be allowed to know the secret shares of other nodes, as this would breach confidentiality. The aim is to hide the actual partial secret shares $S_{p,I}$, while still transporting the combined secret share S_p to node *p*. A shuffling scheme is used to solve this problem. The shuffling scheme is illustrated in Figure 13. From Figure 13, nodes *i* and *j* wish to initialize node *p* with a secret share S*p*. Nodes *i* and *j* agree upon a shuffling factor d_{ij}. The shuffling factor is combined with the partial secret shares $S_{p,i}$ and $S_{p,j}$. The sum of the shuffling factors is null. Therefore this allows for the secret share S_p to be calculated while hiding the secret shares of *i* and *j*. Figure 13 illustrates a system with a threshold of two nodes, to scale this to *k* nodes. Each pair of contributing nodes must decide on a shuffling factor resulting in *k(k-1)/2* shuffling factors which need to be distributed.

This key transport mechanism is described in the following steps:

1. Node *p* broadcast an initial request to a coalition of *k* neighbouring nodes.

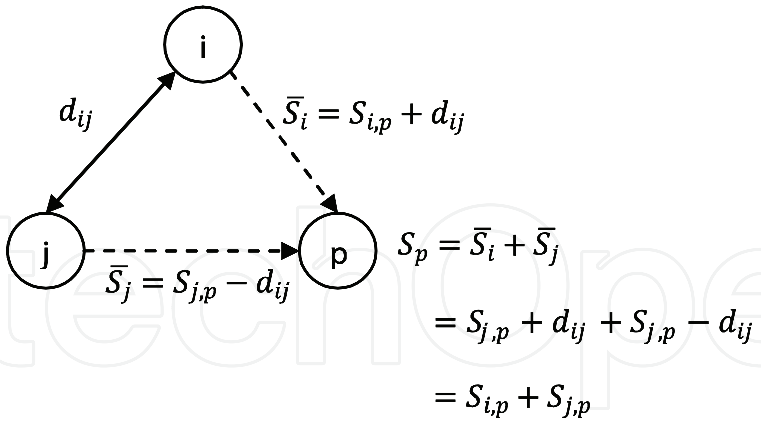

i
d_{ij}
$\overline{S_i} = S_{i,p} + d_{ij}$
j
p
$\overline{S_j} = S_{j,p} - d_{ij}$
$S_p = \overline{S_i} + \overline{S_j}$
$= S_{j,p} + d_{ij} + S_{j,p} - d_{ij}$
$= S_{i,p} + S_{j,p}$

Fig. 13. Shuffling scheme of partial secret sharing

2. The coalition of nodes divides into i and j pairs and agree upon appropriate shuffling factors. An associated public witness $h^{d_{ij}}$ is generated and signed to identify any misbehaviour. The shuffling factor and the witnesses are sent to node p.
3. Node p routes all the shuffling factors and witnesses to the k coalition nodes.
4. Each coalition node j generates the partial secret share $S_{j,p}$ and shuffles it with the shuffling factors received by p such that $\overline{S_{j,p}} = S_{j,p} + \sum_{i=1}^{k} d_{ij}$ and sends $\overline{S_{j,p}}$ to p.
5. Node p verifies the shuffled share values $\overline{S_{j,p}}$ by checking the public witnesses that $h^{\overline{S_{j,p}}} = h^{S_p} \prod_{i=1}^{k} \left(h^{d_{ij}}\right)$. If the verification is successful the shuffled share values are combines such that $S_p = \sum_{i=1}^{k} \overline{S_{p,i}}$.

After the joining node p has been issued with a part of the CA private key, it can perform the services of the CA in the network including certificate renewal and certificate revocation. System maintenance includes the initializing of joining nodes. System maintenance also encompasses the renewal of certificates, certificate revocation and proactive updating of the CA private key shares, therefore protecting against the CA's private key becoming compromised.

d. Share Updating

In a k threshold system, attacks can compromise k nodes over a period of time allow them to impersonate the CA and perform malicious communication attacks. A solution to this is secret share updates by the use of a proactive security method, similar to that used in partial distributed certificate authority methods.

The network will have an operation phase and an update phase where periodic updates will occur of the secret shares of the CA's private key will be updated. During the update phase all nodes participate in the updating procedure. Each node will have an equal probability of initiating the update phase, therefore fairly distributing the load. The secret share update phase following the following steps:

1. The node which is to initiate the update phase requests a coalition of k nodes and generates an update polynomial $f_{update}(x) = b_1x + b_1x^1 + \cdots + b_1x^{k-1}$.
2. Each co-efficient of the polynomial is signed by the coalition CA and flooded through the network such that each node possesses the $f_{update}(x)$ polynomial.
3. Each node i generates its secret update share $\overline{S_i} = f_{update}(ID_i)$ and verifies it by a coalition of k nodes. Each node in the coalition returns a partial update to node i who

combines them to form its update share. This update share is added to the current share and a new updated share of the CA's private key is formed.

The share update procedure provides robust security against multi-point attacks but security comes at a high computational cost.

e. Certificate Renewal

Certificate issuing is assumed to be handled by the off-line TTP, which registers, initialises, and certifies new nodes joining the network. The issue of certificate renewal is performed by the distributed CA in the network. Each nodes certificate is only valid for a specified time period, after which they must renew the certificate before it expires. For successful certificate renewal in a *k* threshold fully distributive system, node *i* must request the renewal of certificate $Cert_i$ from a coalition of *k* nodes. One-hop neighbours are identified as more trust worthy coalition members. Each coalition node then generates a new partial signature and will send it to node *i*. Node *i* then act as a combiner (all nodes may act as combiners in the fully distributive certificate authority scheme) and combines the *k* partial signatures to produce the new certificate $\overline{Cert_i}$ [Luo &Lu, 2000]. In a similar manner, messages are signed by the coalition nodes and form a group signature as described in providing authenticity and security.

f. Certificate Revocation

Certificates can be revoked if nodes are found to be corrupt or compromised. This revocation service assumes that all nodes monitor their one-hop neighbour nodes and are capable of retaining their own certificate revocation list (CRL) [Luo & Lu, 2000]. When a user node identifies a neighbouring node is corrupt, it adds the node in question to its CRL and announces this to all neighbouring nodes. The neighbouring nodes in turn check if this announcement is from a reliable source, i.e. the source is not on the receivers CRL. If the source is reliable, the announced node is marked as suspect. If a threshold of *k's* reliable accusation is made against a single node then the node's certificate is revoked. This procedure allows for compromised nodes to be identified and explicitly quarantined from CA involvement, until such a time as they have become secure again. Implicit revocation is implemented by setting lifetimes for certificates t_{cert}. When the time has expired and the certificate has not been renewed it is implicitly revoked.

g. Analysis

This scheme is a hierarchical model. It is similar to the partially distributed certificate authority scheme. One can see that fully distributive networks possess similar weaknesses to partial distributive networks. Both schemes require prior knowledge and an off-line TTP for the initialization of certificates. The main advantages of the fully distributive scheme are its availability and implement revocation mechanism.

The fully distributive nature of the CA allows for high availability. It does require that each requesting node have *k* one-hop neighbours, which form a CA coalition. The localization of the coalition to the one-hop neighbours avoids transitive trust and reduces network traffic.

One can choose for the threshold parameter *k* to be larger, which will provide a higher level of security. This change requires an attacker to compromise a larger number of nodes in order to obtain the CA's private key. Increased security comes at the cost of availability. This scheme is non-scalable, as it lacks a mechanism that increases the threshold parameter *k,* dynamically, as the network density increases.

As the CA is distributed through the network its availability is greatly increased. However, an increase in availability of the CA requires a greater security and more focus upon the proactive share refreshing scheme. This scheme is a complex and computationally taxing maintenance protocol. It includes the share initialization and share update protocols. The trade-off between security and resources is an important issue in wireless ad hoc networks.

The revocation mechanism allows for explicit and implicit revocation, while the assumption is made that all nodes are computationally capable of monitoring the behaviour of their one-hop neighbours. However, this assumption may not be true for certain ad hoc networks.

3.6 Cluster based model

This solution investigates the Secure Pebblenets [Basagni, 2001], which is a cluster or group based scheme. This solution uses symmetric key cryptography. It is a hierarchical distributive key management system. The focus of this scheme provides group authentication for user nodes, as well as message integrity and confidentiality. Group authentication is achieved by grouping nodes into clusters and treating them with blanket authentication. This solution is suited for planned, long-term distributed ad hoc networks. It is specifically aimed toward networks with low capacity nodes, which lack the resources to perform public key encryption.

a. System overview

This solution requires an initial infrastructure for setup. A secret group identity key k_G is set. This identity provides every node with authentication and integrity. Its key is kept constant for the duration of the network - unless an off-line authority re-initializes the network. k_G is used to generate further keys to provide message confidentiality [Basagni, 2001].

The life of the network is illustrated in Figure 14. The lifetime is divided into time slices, with three phases: the cluster generation phase; the operation phase; and the key update phase. Each time slice consists of these three phases. A network with low processing capacity nodes, authentication is complex and costly. Therefore authentication, confidentiality and integrity are provided for nodal groups or clusters. This maximizes efficiency and minimizes computational cost.

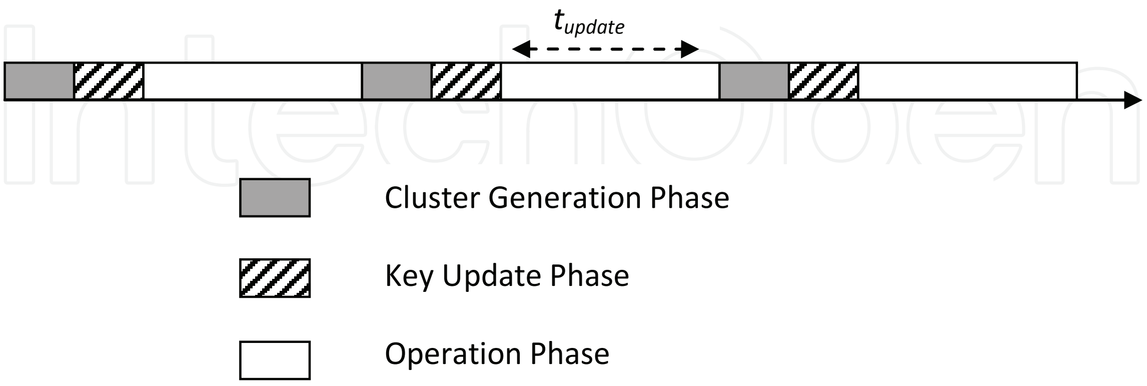


Fig. 14. Phases of the network lifetime

b. Cryptographic keying material

The network uses the following cryptographic keying material to provide message and group confidentiality and authentication:

1. Group identity key k_{GI} is shared prior to network establishment between all network nodes and is used to derive additional keys for security services.
2. Traffic encryption key k_{TEK} is used for symmetric data encryption and is updated during the network lifetime.
3. Cluster key k_C is used for cluster specific communication.
4. Backbone key k_B is used to encrypt communication between cluster heads.
5. Hello key k_H is used between neighbours in cluster generation phase.

The cluster key is generated by the cluster head. The k_{TEK} is randomly generated by the key manager, who is selected in the key update phase. The group identity key is used to derive the backbone and hello keys in the following manner:

$$k_B^0 = k_{GI}$$

$$k_H^i = h\left(k_B^{i-1}\right) = h^i(k_{GI})$$

$$k_B^i = h\left(k_H^{i-1}\right) = h^{i+1}(k_{GI})$$

where k^i represents the key in the i time slice and h^i represents a hash function to the order i. The three phases of operation use the described cryptographic keying material to provide cluster based security in a hierarchical manner.

c. Cluster Generation Phase

During the cluster generation phase, nodes decide to be either cluster heads or cluster members. This decision is based on a variable called weight [Basagni et al, 2001]. Node *i*'s weight w_i is a representation of the node's current capacity status, which is made up of factors such as: battery power, and distance from other nodes etc. The cluster head will manage the group keying services for that cluster. The cluster heads then discover each other and establish a cluster head backbone, which is used to distribute updated traffic encryption key k_{TEK}.

The cluster generation phase follows the following three steps:

1. Nodes share their weights. Each node *i* calculates its weight w_i . It then broadcasts its *id* and w_i to its one-hop neighbours, and encrypts it with the hello key k_H . This provides confidentiality and, along with the group identity key, they provide authentication. The message is as follows.

$$E_{k_H}(w_i|id_i|E_{K_{GI}}(w_i|id_i))$$

2. After receiving the weighted messages from all its neighbours, node *i* will decide if it is a cluster head or cluster member. Once a role has been selected by node *i* it broadcasts its role to its neighbours in the following message.

$$E_{k_H}(w_i|id_i|role|E_{K_{GI}}(w_i|id_i|role))$$

The *role* of node *i* is decided by its weight. The highest weighted node will broadcast a role of *ch*, cluster head, while other nodes will broadcast a role of id_j, where *j* is the identity of the cluster head that node *i* will belong to.

3. The cluster heads are then inter-connected. All cluster members inform their cluster head of any other cluster heads within a three hop radius. The network is effectively segmented and clusters are interconnected by a cluster head backbone, as illustrated in Figure 15.

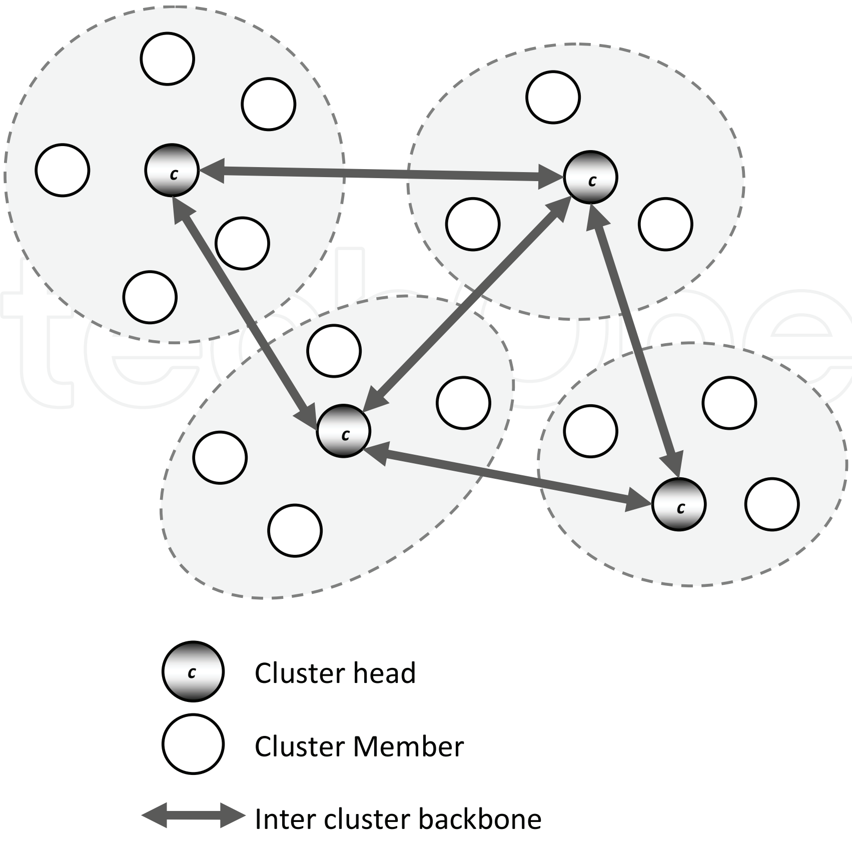


Fig. 15. Segmented network with cluster backbone

d. Operation Phase

During the operational phase, the nodes use the group identity key k_{GI} to authenticate nodes and provide message integrity. The traffic encryption key k_{TEK} is used to encrypt the application data and provide message confidentiality. These services are provided using the cryptographic functions of symmetric encryption algorithms and the one-way hash function [Basagni, 2001].

e. Key Update Phase

The traffic encryption key is updated periodically. This period is measured by an externally set parameter t_{update} (key update period). Updating occurs during the key update phase. Firstly, a key manager is selected from the pool of all the cluster heads. Selection is done by each cluster head, which checks if it is a potential key manager, by comparing its weight with the neighbouring cluster heads. Secondly, an exponential delay period, statistically averaged to Δ, is set aside, as to minimize the risk of multiple nodes becoming key managers [Basagni, 2001]. Thirdly, the cluster head with the highest weight value will arise as the selected key manager. The key managers purpose is to generate a new traffic encryption key k_{TEK} and then distribute this to all the cluster heads, effectively updating the traffic key (which provides message confidentiality). The new k_{TEK} is generated using a secure key generation algorithm. This new traffic key is distributed to the cluster heads securely using the backbone key k_B. The message sent to the cluster heads is:

$$E_{k_B}(w_c|id_c|\overline{k_{TEK}}|E_{K_{GI}}(w_c|id_c|\overline{k_{TEK}}))$$

Once the cluster heads have received the new traffic key this is distributed to the cluster members using the cluster key k_c , which is generated by the cluster head. The message sent to the cluster members is:

$$E_{k_C}(w_c|id_c|\overline{k_{TEK}}|E_{K_{GI}}(w_c|id_c|\overline{k_{TEK}}))$$

These three phases are repeated every network time-slice. The shorter this time-slice, the greater the security obtained. Similarly, this applies to the t_{update} period for the key update phase. However, in this case, it stands that the shorter the update period or time-slice, the more resources are required.

f. Analysis

This scheme is designed for large ad hoc networks, which are made up of nodes with limited processing power and storage capacity. Public key cryptography is unsuited for such a design, as this solution is realized through symmetric key cryptography. This solution requires a TTP to initialise the network nodes with the group identity key k_{GI} and set the parameters, such as the t_{update} time period.

The group identity key, which is distributed to all participating nodes, is required to remain secret throughout the lifetime of the network. In [Basagni, 2001] the authors of the Secure Pebblenets solution propose that nodes have tamper-resistant storage, which securely holds the group identity key. Standard network devices do not have such features and this limits its application for mobile ad hoc networks. If an attacker were to compromise the group identity key, all the nodes in the network would need to be re-initialized with a new group identity key, given by a TTP.

The clustering approach does benefit large ad hoc networks, as routing algorithms for long distances or large networks can become complex and expensive. Cluster based communication allows for packets travelling long distances to travel via the cluster backbone, until they reach their desired neighbourhood or cluster. From there the cluster head can transmit the packets more specifically. This approach reduces security computation and routing complexity in large networks.

A cluster head centralizes the authority in a network. In doing so, it provides a central point of attack for adversaries. Nodes within mobile ad hoc networks have unreliable characteristics because of their mobility and wireless sporadic connectivity. Selecting a reliable cluster head may become a problem in these dynamic networks. Nodes may also refuse to adopt the computational burden of being the cluster head. This is due to resource constraints inherent to mobile ad hoc networks.

Authentication is limited to groups to reduce computational requirements of nodes. It was found that if authentication was to be extended to the individual nodes, it would require the management of $n \times \frac{(n-1)}{2}$ symmetric keys [William, 1999]. Therefore, this solution is not feasible for peer-to-peer communication.

3.7 Proximity-based identification

Smetters et al [Smetters et al, 2002] proposed a solution called demonstrative identification. This solution allows nodes to establish initial trust relationships without prior knowledge or relationship and without the existence of an off-line TTP, which most key management systems assume. This solution uses close proximity channels to establish initial bootstrapping and provides a basis for more complex key establishment. Demonstrative identification approach is designed for spontaneous, small, localized short term ad hoc networks. An example of such a network can be seen in the gathering of people in a coffee shop, where each person wishes to establish temporary communication network, via their PDA's.

a. System Overview

Two nodes desiring to establish a secure communication link, initially engage across a location-limited channel. This channel is separate to the main communication channel, as displayed in Figure 16. Location-limited channels include: infrared; physical contact; and audio etc. Across the location-limited channel pre-authentication information is exchanged. For example, a user with a PDA who wants to communicate with a second user's PDA can use an infrared channel. They can direct the PDA's infrared device towards the second device and an exchange is made. The user can be assured that the pre-authentication information is from the chosen PDA, due to the nature and characteristics of infrared communication.

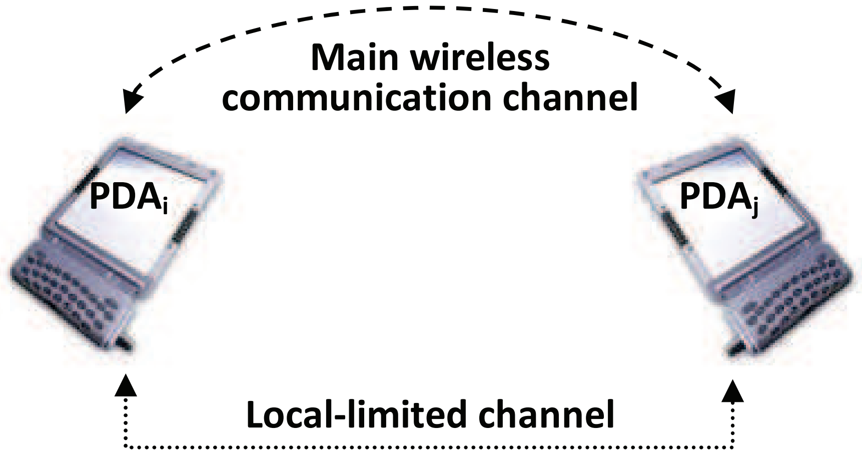


Fig. 16. Proximity based identification with location-limited channel

After the user has exchanged the pre-authentication information, a two-party (for example Diffie-Hellman) or group key exchange scheme can be implemented over the main communication channel. This is done in order to establish the keying material required for secure communication. A limited localized communication channel allows for communication without the existence of an off-line TTP or prior knowledge.

b. Two-Party Key Exchange

The key exchange between communication pair *i* and *j* is explained in the following steps:

1. Nodes *i* and *j* make close proximity contact with each other using a common location-limited channel.
2. Pre-authentication information is exchange across the common location-limited channel. Node *i* sends $h(K_i)$ to node *j* and *j* sends $h(K_j)$ to node *i*, where $h(K_j)$ is the irreversible one-way hash function of a node *j*'s public key.
3. Nodes *i* and *j* now exchange their public keys over the main channel such that *j* receives $\overline{K\imath}$ and *i* receives $\overline{KJ}$. To avoid the impersonation attack which is common to mobile ad hoc networks, the public keys are then authenticated in step 4 using the pre-authentication information from step 2.

4. Authentication is checked using the one-way hash function h and verifies that $h(\overline{K}_\iota)$ = $k(K_i)$ and $h(\overline{K}_J)$ = $k(K_i)$.
5. Upon successful verification, any asymmetric key-exchange protocol can be implemented to allow for nodes i and j to share a secret key.

The two-party key-exchange described above is the basic formulae for demonstrative identification. This protocol can also be applied to heterogeneous nodes, where public key encryption is available to only one of the two communication members. This allows for nodes with limited complexity and computational capacity to participate in pair wise secret key exchange. The procedure for a two-party key exchange, where only one of the members (node i) is the public key competent, is described as follows:

1. Nodes i and j make contact on a location-limited channel, allowing i to send j, $h(K_i)$ and j to send i, $h(S_j)$, where $S_j$is a secret from j.
2. Node i sends j, $\overline{K\ }_\iota$over the main communication channel to realize authentication.
3. Node j authenticates node i's public key, K_i, by verifying that $h(\overline{K\ }_\iota)$ = $h(K_i)$.
4. Upon successful authentication, node j sends $E_{K_i}(S_j)$ to i.
5. $E_{K_i}(\overline{S}_J)$ is decrypted at node i using K_i. $\overline{S}_J$ is then verified by checking that $h(S_j)$ = $h(\overline{S}_J)$. Upon successful verification the two heterogeneous parties share a secret S_j , which can be used to establish secure communication keying material.

c. Analysis

This solution allows for a fully self-configured ad hoc network, as the initial trust establishment phase does not require the assistance of an off-line TTP. Users realize the initial trust relationship by localized communication. For example, a user with a PDA would point its PDA to another PDA to automatically exchange authentication information and establish a secure communication line.

This solution requires that nodes are equipped with location-limited communication devices. Examples of these devices are: infrared, audio or a wired link. This requirement limits the network participants to those possessing specific peripherals. The assumption is made that most portable wireless devices are equipped with some type of localized communication medium, such as infrared.

The location-limited pre-authentication exchange realizes demonstrative identification [Smetters et al, 2002]. It only allows key-exchange to occur in a localized manner, where nodes are in close proximity to each other. As a result, this solution is not suited to large networks, but it is best suited to small spontaneous networks. A solution presented by Capkun [Capkun et al, 2006] extends the self-issued certificate chaining approach as it implements a demonstrative identification approach in a PGP based network. Capkun's proposal uses location-limited communication to establish initial trust and relies upon mobility to distribute this trust in large networks. Such a proposal allows for demonstrative identification to be implemented in large to moderate networks.

More recently, the Amigo proximity-based authentication system proposed by Scannell et al [Scannell et al, 2009], uses shared radio environment evidences as proof of physical proximity to authenticate localized mobile communication nodes.

3.8 Self issued certificate chaining

A PGP-based security solution for ad hoc networks is proposed by Capkun and Hubaux [Capkun et al, 2003] [Hubaux et al, 2001]. This solution uses a certificate chaining approach.

It outlines a fully self-organized public key management system that allows users to: generate their public-private key pairs; issue certificates; and perform authentication, without the presence of an off-line trusted third party. Capkun and Hubaux focus on the key management and key distribution system. Without the need of prior relationships or an organizational TTP member, this solution is best suited to spontaneous ad hoc networks. However, due to its complex initialization phase it is not suited for small short-term networks.

a. System Overview

Public keys (*K*) and certificates are modelled as direct graphs *G(V,E)* where vertices, *V*, represent the public keys and the edges, *E*, represent a certificate between two vertices. The self-organized system proposed by Capkun and Hubaux [Capkun et al, 2003] [Hubaux et al, 2001] differs from PGP in that it relies on the users to store and distribute the certificates in a self-issued manner. Each user node carries a certificate memory, consisting of certificates limited to local neighbourhood. For a user to authenticate and certify another user's public key, a certificate chain is first found between the two users, by combining the users' certificate memory. Figure 17 illustrates a situation where node *u* and *v* request secure communication [Capkun et al, 2003]. Node *u* is required to verify the authenticity of the public key K_v for corresponding to node *v*. To do so nodes *u* and *v* combine their certificate memories to find a certificate chain or path between K_u and K_v , which is made up of valid public key certificates shared between the two communicating nodes.

The fully self-organized public key management system can be broken into four procedures of analysis, as follows:

- Public/private key creation
- Certificate exchange
- Authentication
- Certificate revocation
- Load sharing

During the initialization phase, the public-private keys are created and distributed with a certificate exchange procedure. Secure communication is realized and impersonation attacks are thwarted by the authentication of the available certificates. The certificate revocation protocol is outlined in order to maintain security and exclude malicious users. Optimization is implemented by a load sharing protocol that ensures fair distribution of the work load and prevents selfish nodes in a network.

Initialization phase is executed in a four step procedure which establishes trust in the network:

1. The user creates their own public/private key pair
2. The user issues public key certificates (vertices) based on the knowledge of the other public keys.
3. The user performs certificate exchange and collecting certificates, and creates a non-updated certificate repository.
4. The user constructs an updated certificate repository, modelled as a graph G_u. This is done by communicating with certificate graph neighbours or by a second method of applying the repository construction algorithm to the non-updated certificate repository.

After initialization is complete, authentication between two users can take place, through certificate chaining. Each step is explained in more detail below.

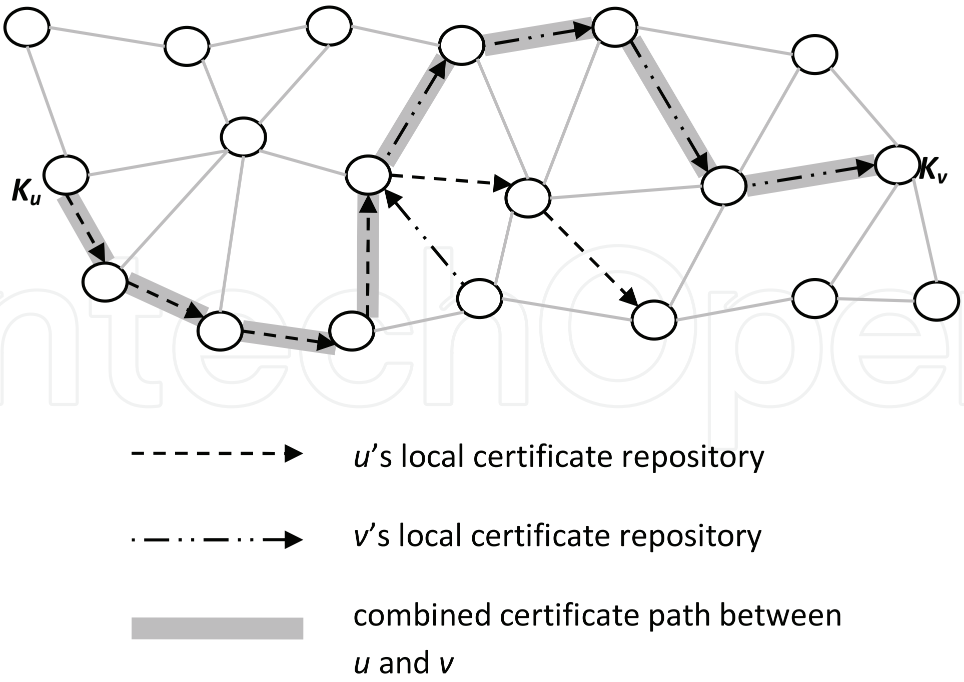

Ku
Kv
u's local certificate repository
v's local certificate repository
combined certificate path between u and v

Fig. 17. A certificate chain or path between public keys K_u and K_v

b. Public/private Key Creation

Public and private keys for users are created locally. Public key certificates are issued by the user. If the user *u* believes that a public key K_v belongs to *v*, then the user *u* can issue a public key binding K_v to user *v*, by the signature of *u*. This certificate has an expiry time T_v. A periodic update may be issued which simply extends expiry time T_v. The reason for trust is not identified but assumed, for example through a physical side channel.

c. Certificate exchange

The certificate repositories are created automatically by exchanging certificates. A user *u* has two certificate repositories: an update certificate repository G_u and a non-updated certificate repository G_u^N. All certificates are stored twice, as when a certificate is issued, it is stored in both the certificate issuer *u* and certificate owner *v*'s repository. Therefore, initially each certificate repository has only the certificates it has issued and those that have been issued to it. Certificates are exchange periodically. Each node periodically polls its physical neighbour for certificates.

A certificate exchange is performed by the following procedure:

1. Node *u* broadcasts G_u and G_u^N to its physical neighbours. The broadcast contains only identities (hash values).
2. Neighbours reply with identities of their update repository G and non-update repository G^N.
3. Node *u* crosschecks the received sub-graphs and its sub-graphs for any additions.
4. Node *u* requests those certificates it does not hold.

After the initial convergence phase, all the certificates of the nodes are stored by all users. As a result, users' non-update repositories are created. After this phase the nodes exchange only new certificates at a rate of T_{CE}, which represents the time for a certificate to be exchanged throughout the network. Note that certificate expiration times are not considered thus far.

d. Construction of updated certificate repositories

The exchange of certificates provides an incomplete view of the graph and allows each node to create its own non-updated certificate repository. The updated repository G_u will consist of certificates which user u keeps updated. There are two approaches in this creation:

1. Apply algorithm A to G_u^N which results in G_u, and validity of each certificate is checked.
2. Communicate with certificate graph neighbours only.

The maximum degree algorithm is an algorithm A proposed by [Capkun et al, 2003] which is applied to the non-update repository G_u^N to create the update repository G_u in [Capkun et al, 2003] [Hubaux et al, 2001]. The algorithm selects a sub-graph that consists of two logically distinct paths: the out-bound path and the in-bound path, which are made up of outgoing edges and incoming edges, respectively. The selection of G_u's out-bound path is done in multiply rounds in the following manner [Capkun et al, 2003] [Hubaux et al, 2001]:

1. Each round runs from vertex K_{vert}, starting with vertex K_u.
2. User u requests the outgoing edge list of vertex K_{vert}. This is possible as every vertex stores this list locally.
3. An outgoing edge (with its terminating vertex z) is selected from the list in 2. Selection is based on the highest number of shortcuts of the terminating vertex z. Where a shortcut is defined as an edge, and removed, the shortest indirect path between the nodes, previously connected by that edge, becomes larger than two. User u can determine its number of shortcuts by gathering information about the outgoing and incoming edges of its adjacent users.
4. The selected vertex z is added to a set N_{out} of vertices selected, thus far. This is done to ensure that the selected out-bound paths are disjointed.
5. The round is finished and now the terminating vertex z becomes K_{vert} and a new round begins, starting from step 1.

The in-bound path selection is done in a similar way:

1. Each round runs from vertex K_{vert} , starting with vertex K_u.
2. User u requests the incoming edge list of vertex K_{vert}. Every vertex stores this list locally. Therefore, this step requires that each user be notified whenever another user issues a certificate to that user.
3. An incoming edge (with its originating vertex y) is selected from the list in 2. Selection is based on the highest number of shortcuts of the originating vertex y.
4. The selected vertex y is added to a set N_{in} of vertices selected so far, to ensure that the selected in-bound paths are disjointed.
5. The round is finish and now the originating vertex y becomes K_{vert} and a new round begins, starting from step 1.

The update repository is the union of the in-bound sub-graph and out-bound sub-graph. The pure method will operate on a single round. However, it is extended so the update repository consists of several vertex disjoint out-bound and vertex disjoint in-bound paths. The final sub-graph is star-like information.

e. Authentication

When initialization is complete, the user is prepared to perform authentication. Authentication is preformed between users u and v with public keys K_u and K_v respectively, as follows:

Firstly, user u and user v merge their update certificate repository (G_u and G_v) to find a certificate chain between u and v. User u then looks for a path in G_u and G_v. Validity and correctness checks are done to all certificates in the discovered path. Validity, checks that the certificates are not revoked. Correctness, checks the certificates contain the correct user-key bindings.

If no certificate chain is found, user *u* combines its two repositories of the updated and non-updated certificates to find a chain. User *u* searches for a path in G_u and G_u^N. If a chain is found, then *u* requests the updates of the expired certificates. Subsequently, the validity and correctness checks are made.

If there is still no certificate chain found between K_u and K_v then authentication is aborted. During authentication nodes that are one-hop physical neighbours (also known as helper nodes) are given precedence as to maximize performance. When a path is found, the certificates (edges) along this path are then used by user *u* to authenticate K_v.

f. Certificate revocation

Certificates are revoked when it is believed that the user-key binding is no longer valid. If a user believes his own private key is compromised then he can revoke his public key certificate binding. This is done in two ways, explicitly and implicitly:

1. Explicitly, a user *u* would revoke a certificate issued by *u,* by broadcasting a revoke statement broadcast to its G_u nodes. The certificate exchange scheme allows for this revoke to reach all other nodes at a time delay of T_{CE}.
2. Implicit revocation is based on the expiration of certificates. Certificates are valid for a given time T_v after which they must be updated.

This allows for comprised certificates and private keys, to be dealt with explicitly, and provides a higher level of confidence by implicitly maintaining validity.

The fully distributive nature of this scheme means every certificate is stored at each node allowing for nodes to cross-check conflict and detects inconsistent certificates.

To combat false certificate bindings the following two procedures are taken:

1. If a certificate is received which doesn't exist in G_u or G_u^N then it and the issuer are labelled *unspecified* until a period T_p where $T_p > T_{CE}$ where after if no conflicting certificates are received then it is marked *non-conflicting*. This does not prevent against Sybil attacks though.
2. If a certificate conflict is found where a user *u* has two certificate bindings (v,K_v) and (v,K'_v). Both certificates and the certificates that certified them are labelled as *conflicting*. To resolve such a conflict, validity of certificates is first checked with their issuers. If validity status remains true, then *u* will try to find chains of non-conflicting valid certificates to public keys K_v and K'_v. Confidence values are calculated based on the number and length of chains, and values compared to compute the correctness of the bindings. If no decision is made these bindings are labelled as *conflicting* and the node waits for more information to resolve the conflict.

In this case, a confidence algorithm is not identified but assumed. This conflict resolution mechanism can be further used: to evaluate trust in users; to issue correct certificates; and to detect malicious users.

g. Load Sharing

For an update to occur nodes contact the issuer of the certificates that they store. This approach is not efficient because one certificate issuer could be overloaded and unable to handle the computational work load. Simple load sharing is implemented which allows for relief. Each node *u* provides updates to up to *s* other nodes, where *s* is equal to size of *u*'s updated repository. After which node *u* has provided *s* updates, it replies to update requests with a list of nodes that get updates directly from *u*. The requesting node then randomly selects a node from *u*'s list and requests its update from that node.

h. Analysis

The self organized, self certificate issuing trust model is a web of trust type model inheriting PGP characteristics and applying them to an ad hoc network environment. In a similar way that PGP [Zhou & Hass, 1999] realizes trust, the certificate chaining approach is used to create chains of hierarchical trust between users. The main difference between PGP and the certificate chaining solution is that PGP stores certificates in a centralized manner, and this scheme decentralizes this procedure through local certificate repositories.

The main advantage of this scheme is that it is fully self-organized and does not require the presence of a TTP. Trust is established in a self-organised manner with self-certificate being issued by the nodes themselves. The initial phase requires nodes to interact and establish trust. Trust relationship can take time to establish. Therefore, in the early stages of the network, an initial time delay can be expected limiting the effectiveness of communication. For this reason, this network is not suited for short term mobile ad hoc network. An example of this shortcoming is illustrated in Figure 18, where node *A* wants to communicate with node *B*. At the early stage of the network only *D* and *C* have issued certificates and as a result no certificate chain exists between *A* and *B*. Only once the intermediate nodes have issued certificates will a certificate chain between *A* and *B* be possible.

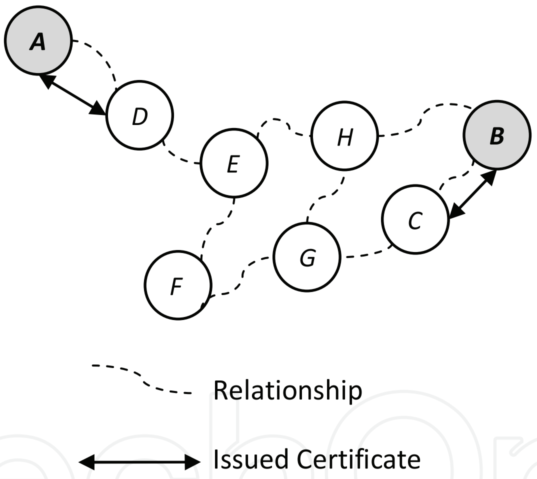


Fig. 18. Initial phase delay problem

The use of certificate chains is identified as vulnerable, because a chain of trust is 'only as strong as its weakest link'. A PGP hierarchical trust model is adopted that assumes transitive trust. This web-of-trust based approach allows for more flexibility than the other certificate approaches. However, a no central administration is present to enforce policy and trust assessment. Therefore, because of this lack of structure, it is more prone to attacks by malicious nodes. This solution is best suited to open mobile ad hoc networks, but may not be suited to applications where high degrees of security is required [Davis, 2004], like closed military mobile ad hoc networks.

This self-organized scheme is fully distributive which would result in a certificate updated to be computationally taxing. Certificate update repositories and load sharing relieve this expense. However, a better load balancing data management schemes can be introduced to further relieve the load [Hubaux et al, 2001].

The maximum degree algorithm *A* (or Shortcut Hunter Algorithm) is implemented to maximise effectiveness and optimise the update procedure. This proposal has been tested on PGP trust graphs. Nevertheless, an ad hoc network does not have the privilege of every node having public knowledge of all the certificates available. Step 3 of the maximum degree algorithm requires that an edge is selected from *vert* to *z* , where *z* is the vertex with the highest number of shortcuts. To determine *z* knowledge of the surrounding trust graph is required, which may not be available to all ad hoc network members.

One of the main disadvantages of a fully self-organized model is that nodes can adopt as many identities as they have resources, in order to support further steps which need to be taken to protect this solution from Sybil and impersonation type attacks [Capkun et al, 2003].

3.9 Discussion and summary

The solutions presented in this section give a summary of the work related to key management in mobile ad hoc networks. The solutions differ considerably in requirements, complexity and functionality. Each solution is suited for different types of ad hoc network environments. Criteria which these key management solutions can be grouped or differentiated included:

- Pre-configuration: *Planned vs Spontaneous*

This describes the pre-requisites and assumptions that are made for the nodes participating or joining the network. If an ad hoc network is planned then nodes can be assumed to have some pre-configured information, for example: initial shared secret; certificate; or authenticated identification. If the network is spontaneous then nodes have no prior security relationships or initial data assumptions. Pure ad hoc networks are more spontaneous allowing for nodes to join and leave the network without complex pre-configurations and assumptions made.

- Network Area: *Local vs Distributive*

This describes the area or space in which the key management scheme is operating. The physical topology of the network would result in more close proximity interaction or more multi-hop distributive interaction. A localized area is a network in which nodes come within a close proximity range of each other, such as in a classroom. A distributive area is a network where nodes are located some distance apart with little possibility of physical interaction. Certain key management schemes do not function in a distributive network area.

- Network Duration: *Short Term vs Long Term*

The duration of the network can dictate the initialization period of the key management scheme. For short term ad hoc networks, a group of nodes establish communication for a short time period and may never come into contact again. Short term ad hoc networks require speedy initialization and require communication to be available at the start of the network, without an initial period of weakened or delayed secure communication. Long term ad hoc networks consist of nodes that plan to be part of a network and in relationship with other nodes for a longer time period. Furthermore, nodes retain information and relationships with other nodes even when they leave the network. Long term ad hoc networks require more complex trust establishment.

- Off-line TTP Involvement

Ad hoc networks are characterized by their lack of infrastructure. Key management scheme often rely on an off-line trusted third party (TTP) for initialization and operational security. The extent of the off-line TTP involvement describes the self-organized nature of the network. Ideally, an ad hoc network has no off-line TTP involvement at the initialization or operational stages.

A summary of the presented key management solutions given in Table-1 with respects to the criteria discussed above. The off-line TTP model relies on an external TTP to establish and maintain security. This model is suited for networks which have available fixed infrastructure and will therefore have limited mobility. The partially and fully distributive CA solutions are similar using threshold cryptography, as they distribute the hierarchical trust of a certificate authority. They are suited to large planned ad hoc networks like military battlefield networks or disaster area networks. The Secure Pebblenet scheme is a cluster based model which is ideal for hierarchical group-oriented ad hoc networks where all nodes are distributed in a large network area and nodes have limited resources. An application of this cluster based approach is sensor networks.

The Self-Issued Certificate model or certificate chaining model uses a localized PGP web of trust approach. Its self-organized nature makes this solution most suited to spontaneous networks, such as peer-to-peer communication in a classroom or coffee shop. The proximity-based identification solution is suited to localized networks. Its greatest advantage is that it requires no prior knowledge to establish trust. The proximity-based identification method is, used in Capkun's mobility based approach, uses mobility of nodes to establish initial trust relationships across a large network.

This section shows that many of the solutions presented have issues which need to be resolved. Key management is an integral part of providing security and, as identified in Section-1, the routing layer is the focus of attack for adversaries. If these MANETS are to be recognized as secure, then mobile ad hoc network's security mechanism must strive to provide security on the routing and application layer.

	Pre-Configuration	*Network Duration*	*Network Area*	*Off-line TTP Involvement*
Off-line TTP Model	Planned	Long-term	Distributive	Full
Partially Distributed CA	Planned	Long-term	Distributive	Initialization
Fully Distributed CA	Planned	Long-term	Distributive	Initialization
Self Issued Certificates	Spontaneous	Long-term	Distributive	None
Cluster based Model	Planned	Long-term	Distributive	Initialization
Proximity-base Identification	Spontaneous	Short-term	Localized	None

Table 1. Summary of Key Management Solutions

4. References

[Abdul-Rahman, 1997] A. Abdul-Rahman, "The PGP trust model," *EDI-Forum: The Journal of Electronic Commerce,* vol. 10, pp. 27-31, 1997.

[Aram et al, 2003] K. Aram, K. Jonathan, and A. A. William, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Awerbuch et al, 2002] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of the 1st ACM workshop on Wireless security* Atlanta, GA, USA: ACM, 2002.

[Basagni et al, 2001] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking \&amp; computing* Long Beach, CA, USA: ACM, 2001.

[Bruce, 2003] S. Bruce, *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*: Springer-Verlag New York, Inc., 2003.

[Capkun et al., 2003] S. Capkun, L. Butty, and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* vol. 2, pp. 52-64, 2003.

[Capkun et al, 2006] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Mobility Helps Peer-to-Peer Security," *IEEE Transactions on Mobile Computing,* vol. 5, pp. 43-51, 2006.

[Chor et al, 1985] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)," *proc. 26th IEEE Annual Symposium on Foundations of Computer Science,* October, 21-23 1985.

[Davis, 2004] C. R. Davis, "A localized trust management scheme for ad hoc networks. ," *In: 3rd International Conference on Networking (ICN'04),* pp. 671–675, 2004.

[Desmendt & Jajodia, 1997] Y. Desmedt and S. Jajodia, "Redistributing Secret Shares to New Access Structures and Its Applications," Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Technical ReportJuly 1997.

[Douceur, 2002] J. R. Douceur, "The Sybil Attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*: Springer-Verlag, 2002.

[Eschenauer & Gligor, 2002] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *proc. 9th ACM Conf. on Computer and Communication Security (ACM CCS'02),* November, 17-21 2002.

[Frankel et al, 1997] Y. Frankel, P. Gemmell, D. MacKenzie, and M. Yung, "Optimal resilience proactive public key cryptosystems," *proc. 38th Annual Symposium on Foundations of Computer Science (FOCS '97),* October, 19-22 1997.

[Haas et al, 2002] Haas J.D.Z., Liang B., P. Papadimitatos and S. Sajama, "Wireless ad hoc networks," in *Encyclopedia of Telecommunications* J. W. John Proakis, Ed., 2002.

[Hashmi & Brooke, 2008] Hashmi S. and J. Brooke, "Authentication Mechanisms for Mobile Ad-Hoc Networks and Resistance to Sybil Attack," in *Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies - Volume 00*: IEEE Computer Society, 2008.

[Herzberg et al, 1995] Herzberg A., S. Jaracki, H. Krawczyk, and M. Yung, "Proactive Secret Sharing Or: How to Cope With Perpetual Leakage," *proc. Advances in Cryptology - CRYPTO '95,* 1995.

[Herzberg et al, 1997] Herzberg A., M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive Public Key and Signature Systems," *proc. 4th ACM Conf. on Computer and communications security,* April, 1-4 1997.

[Hu et al, 2003a] Hu Y.C., A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proceedings of the 2nd ACM workshop on Wireless security* San Diego, CA, USA: ACM, 2003.

[Hubaux et al, 2001] Hubaux J.-P., L. Butty, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking \&amp; computing* Long Beach, CA, USA: ACM, 2001.

[Jarecki, 1995] Jarecki S., "Proactive secret sharing and public key cryptosystems," Massachusetts Institute of Technology (MIT), 1995.

[Jameson, 2008] Jameson H., "Secure Military Networks: The war without weapons," 2008.

[Karl & Rauscher, 2001] Karl W.C., F. Rauscher, "Wireless Emergency Rescue Team (WRET) Final Report for the September 11, 2001 New York City World Trade Center Terrorist Attack," 2001.

[Lidong & Zygmunt, 1999] Lidong Z. and H. Zygmunt, "Securing Ad Hoc Networks," Cornell University1999.

[Luo & Lu, 2000] Luo H. and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," Computer Science Department, University of California, Technical ReportOctober 2000.

[Luo et al, 2002] Luo H., P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing Ad Hoc Wireless Networks," *proc. Seventh International Symposium on Computers and Communications (ISCC'02),* July 1-4 2002.

[Menezes et al, 1996b] Menezes A.J., S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*: CRC Press, Inc., 1996.

[Molva & Michardi, 2003] Molva R. and P. Michiardi, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks (extended abstract)," in *proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, 2003.

[Papadimitratos & Hass, 2003] Papadimitratos P. and Z. J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Perkins & Bhagwat, 1994] Perkins C.E. and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234-244, 1994.

[Perkins et al, 2003] Perkins C., E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*: RFC Editor, 2003.

[Qian & Li, 2007] Qian L., N. Song, and X. Li, "Detection of wormhole attacks in multi-path routed wireless ad hoc networks: a statistical analysis approach," *J. Netw. Comput. Appl.*, vol. 30, pp. 308-330, 2007.

[Raya & Hubaux, 2005] Raya M. and J. P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*, 2005.

[Salem et al, 2005] Salem N.B., L. Buttyan, J.-P. Hubaux, and M. Jakobsson, "Node Cooperation in Hybrid Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* 2005.

[Scannell et al, 2009] Scannell A., A. Varshavsky, A. LaMarca, and E. D. Lara, "Proximity-based authentication of mobile devices," *Int. J. Secur. Netw.,* vol. 4, pp. 4-16, 2009.

[Shamir, 1979] Shamir A., "How to share a secret," *Communications of the ACM,* vol. 22, pp. 612-613, 1979.

[Smetters et al, 2002] Smetters D.B., D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks," 2002.

[Stalling, 2002] Stallings W., *Cryptography and Network Security: Principles and Practice*: Pearson Education, 2002.

[Stalling, 2003] Stallings W., *Cryptography and Network Security: Principles and Practices*: Prentice Hall, 2003.

[Steiner et al, 1996] Steiner M., G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Groups," in *proc. Third ACM Conf. on Computer and Communication Security*, 1996.

[Talzi et al, 2007] Talzi I., A. Hasler, S. Gruber, and C. Tschudin, "PermaSense: investigating permafrost with a WSN in the Swiss Alps," in *Proceedings of the 4th workshop on Embedded networked sensors* Cork, Ireland: ACM, 2007.

[Tseng et al, 2003] Tseng C.Y., P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks* Fairfax, Virginia: ACM, 2003.

[Van der Merwe &Dawoud, 2004] Van der Merwe J., D. Dawoud, and S. McDonald, "A Proactively Secure Threshold-multisignature Scheme based on Publicly Verifiable Distributed Key Generation and Publicly Verifiable Secret Redistribution," *IEEE Transactions on Parallel and Distributed Systems,* 2004.

[Van der Merwe &Dawoud, 2005] Van der Merwe J., D. Dawoud, and S. McDonald, "Fully Self-Organized Peer-to-Peer Key Management for Mobile Ad Hoc Networks," *proc. ACM Workshop on Wireless Security (WiSe'05),* September, 2 2005.

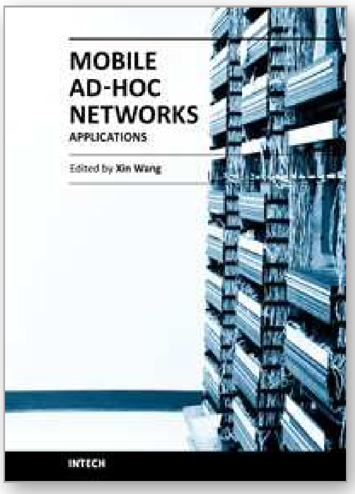
[Verma et al, 2001] Verma R., D. O'Mahony, and H. Tewari, "NTM- Progressive Trust Negotiation in Ad Hoc Networks," 2001.

[William, 1999] William S., *Cryptography and network security (2nd ed.): principles and practice*: Prentice-Hall, Inc., 1999.

[Yi & Kravets, 2001] Yi S. and R. Kravets, "Practical PKI for Ad Hoc Wireless Networks," Department of Computer Science, University of Illinois, Technical ReportAugust 2001.

[Yi & Kravets, 2003] Yi S. and R. Kravets, "MOCA: Mobile certificate authority for wireless ad hoc networks," in *proc. of the 2nd Annual PKI Research Workshop (PKI 2003)*, 2003.

[Zhou & Hass, 1999] Zhou L. and Haas Z.J., "Securing Ad Hoc Networks," *IEEE Network: special issue on network security,* vol. 13, pp. 24-30, 1999.


MOBILE
AD-HOC
NETWORKS
APPLICATIONS
Edited by Xin Wang
INTECH

Mobile Ad-Hoc Networks: Applications
Edited by Prof. Xin Wang

ISBN 978-953-307-416-0
Hard cover, 514 pages
Publisher InTech
Published online 30, January, 2011
Published in print edition January, 2011

Being infrastructure-less and without central administration control, wireless ad-hoc networking is playing a more and more important role in extending the coverage of traditional wireless infrastructure (cellular networks, wireless LAN, etc). This book includes state-of the-art techniques and solutions for wireless ad-hoc networks. It focuses on the following topics in ad-hoc networks: vehicular ad-hoc networks, security and caching, TCP in ad-hoc networks and emerging applications. It is targeted to provide network engineers and researchers with design guidelines for large scale wireless ad hoc networks.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821