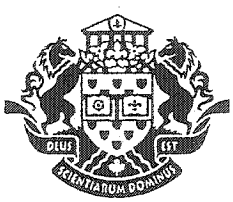


Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Mohamad Jumaid BOHIO

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. A. Sc. (Electrical Engineering)

GRADE - DEGREE

Department of Electrical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Identity-based Security Solutions for Mobile Adhoc Networks

A. Miri

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

M. El-Tanany

A. Karmouch

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

IDENTITY-BASED SECURITY SOLUTIONS FOR MOBILE AD HOC NETWORKS

by

Muhammad Junaid Bohio

A Thesis
submitted to the Faculty of
Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the
Degree of Master of Applied Sciences
in
Electrical Engineering

Master's Thesis

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
August 2004

Your file Votre référence
ISBN: 0-494-01421-0
Our file Notre référence
ISBN: 0-494-01421-0

Abstract

In this thesis, we propose security solutions for pairwise and broadcast communication in mobile ad hoc networks. We use identity-based keys that do not require certificates and that simplify key management. We use pairwise symmetric keys that are computed non-interactively by the nodes, which reduces communication overhead. Our pairwise-key management protocol requires a minimum number of keys, $O(N)$, to be generated by the third party as compared to the conventional pairwise schemes with $O(N^2)$. We also propose an efficient identity-based signature scheme for an authenticated broadcast protocol, a collision-free method for computing broadcast keys, a key escrow free scheme and a signcryption scheme as an alternative to our signature-encryption algorithm for broadcast protocol. Since in the group key distribution in mobile ad hoc networks there is the possibility of packet loss due to the mobility of users and wireless channels, we also propose self-healing, mutual-healing and some optimization techniques for the recovery of lost session keys.

Acknowledgments

First of all, I would thank God Almighty for all His help and blessings that enabled me to achieve my goals. I would then say thanks to my supervisor, Professor Ali Miri, for being so kind and helpful throughout my research. I appreciate his guidance and very useful comments that helped me a lot to improve the security solutions described in this thesis. I learned from him how to look critically at any security protocol, which I think is a valuable asset for me.

I would also thank my colleagues at the CLiCC lab. Their comments, during my presentations, helped me to explain my work in a better way, and take any precautions against any possible attacks on these solutions.

Table of Contents

List of Figures

Chapter 1

Introduction

In this chapter, we will give an overview of the research work described in this thesis, outline of the next chapters, list of the contributions and publications resulting from this work, and discuss the implementation of the proposed solutions.

1.1 Overview

In the recent years, mobile ad hoc networks have been a very active research area, mainly because of the typical nature of this environment and new challenges emerged thereof. The recent popularity of wireless devices is the driving force behind the research activities in this area. In addition to several other issues, security is one major challenge requiring appropriate solutions for this environment. There has been extensive research on the security issues in wired network environments, but those solutions are difficult to implement in the wireless environment (e.g. due to limited bandwidth), and even more difficult in the wireless mobile ad hoc environment for several reasons. Those reasons include: lack of central infrastructure and reliance on the participating users for relaying and routing of the messages, mobility of nodes causing frequent changes in the network layout, use of small devices that are vulnerable to theft, capture or compromise, devices used have limited memory, computational and battery power, and many other reasons that will be discussed in the next chapter.

Considering the above challenges, we realized the need to look for solutions other than the conventional cryptographic techniques. Identity-based cryptography, being a new direction in the security area, offers a potential solution. We found very promising features of this cryptography that can work well under the restrictions of the mobile ad hoc network environment. In this thesis, we present our proposed solutions for mobile ad hoc networks that are mostly founded on the identity-based cryptography. We have proposed authenticated pairwise and broadcast protocols in general, and several integral tools such as a signature scheme, key generation and distribution protocols, and sign-cryption schemes, in particular. In addition to that, we have also proposed solutions for lost key recovery in a group key distribution protocol. Over all, we consider the essential issues of confidentiality, authentication, non-repudiation, and the integrity. We also address the key management issue, but it is not completely distributive. For identity-based key generation and distribution, we describe a method for using multiple Private Key Generation authorities (PKGs), instead of a centralized key generation authority. In key distribution in mobile ad hoc network, there is the possibility of packet loss due to mobility and wireless channels, we have also proposed self-healing and mutual healing solutions for lost key recovery. More details about these proposed solutions are listed in the next section.

1.2 Contributions

In our research work into security solutions for mobile ad hoc network, we have made following contributions:

- We have proposed an authenticated pairwise communication protocol that reduces the key generation and distribution overhead for the Trusted Authority (TA) to $O(N)$, as opposed to the conventional schemes using $O(N^2)$ overhead. It also reduces the storage requirement for the users in a manor similar to the source routing technique in ad hoc networks.

- We have proposed an authenticated broadcasting protocol that also addresses the

issue of confidentiality and integrity.

  - For authentication and non-repudiation in the broadcast protocol, we have proposed an identity-based signature scheme that is comparable to the most efficient signature scheme proposed in the literature.

  - We have developed an identity-based signcryption scheme that provides properties of both signature and encryption in one algorithm, and ensures implicit control of the TA over the broadcast keys generated by the nodes themselves.

  - For the broadcast key computation by the nodes, we have proposed a novel technique for computing collision-free broadcasting keys. Given the identities of the broadcast group, such key computation by all nodes is non-interactive, and thus does not require additional communication among nodes.

- We have proposed a key escrow free scheme for both pairwise and broadcast protocols. Since the conventional identity-based cryptosystem suffers from key escrow problems, the proposed key escrow free solution provides an alternative that ensures the privacy of the users from the key issuing authority.

- For the group key distribution in the network, we have developed self-healing, mutual-healing and optimization techniques for the key distribution protocol of Subset Difference method. Such healing properties in the key distribution will enable nodes to recover lost session keys without contacting the key issuing authority, thereby reducing communication overhead and avoid a bottle-neck at the distribution authority. The proposed optimization techniques will help to control the message size and encryption/decryption operations.

- We have implemented the software for the above mentioned solutions based on the identity-based cryptography. We used the MIRACL cryptographic library to implement the proposed solutions on 512-bit and 1024-bit numbers. Since MIRACL also provides implementation of the AES symmetric cipher (in addition to the identity-based encryption), we have implemented the algorithm for computing pairwise shared keys from identity-based cryptography and then used it with AES.

Most of the above proposed solutions have been published in [37, 35, 36] and [38].

1.3 Thesis outline

The next chapters of this thesis are organized as follows. Chapter 2 consists of a survey of a wide range of security problems in mobile ad hoc networks and some proposed solutions. In the conclusion of that chapter, we mention the problems on which we have worked in our research. In chapter 3, we give an introduction to identity-based cryptography as most of our solutions are based on this new area of cryptography, and give some technical details that are required to explain the proposed solutions. Chapter 4 consists of proposed solutions for authenticated pairwise and broadcast protocols. Chapter 5 describes our solutions to self-healing, mutual-healing and optimization techniques for key distribution in ad hoc networks; and chapter 6 is the conclusion and outlines future work and possible extensions of this research.

Chapter 2

Security in Wireless Mobile Ad Hoc Networks

In this chapter, we give a brief introduction to mobile ad hoc networks and list security challenges due to the nature of this typical network. Later, we discuss security threats in the perspective of ad hoc network environments, desired security mechanisms, and possible solutions that have been proposed in the literature. Overall, in this chapter we give a broad perspective of security problems in ad hoc networks that include availability, key management, authentication, node misbehavior, intrusion detection, confidentiality, and so on. Finally, we list the problems and issues which are considered in this thesis and the extent to which they are resolved.

2.1 Introduction

A mobile ad hoc network is a self-configurable, autonomous, and infrastructureless wireless network. Users or nodes in the network can be moving or static; some nodes could be joining the network while others are leaving; some nodes may have very limited resources (e.g. nodes with hand-held device) as compared to others, and so on. Thus, it has a quite dynamic structure and variable node status (location, number of users in the network, active or passive nodes, etc.). Since the network is infrastructureless, that is the network can be promptly deployed without relying on any existing infrastructure such as

base stations for wireless cellular networks, it requires the cooperation of the participating nodes to carry out network operations. Such operations include forwarding and routing of the packets from other nodes, sharing network information (e.g. routing updates), authorizing new nodes in the network, and, in some protocols, monitoring of the behavior of neighbouring nodes.

The primary applications of mobile ad hoc networks include military operations, emergency and disaster recovery operations where the environments are hostile and the operations are security-sensitive requiring fast and reliable deployment. However, due to the increasing popularity of wireless devices, the interest in mobile ad hoc networks has been extended to civilian life such as on-the-fly setups for conferencing, mobile internet, academic institutions, etc.

In the mobile ad hoc network research area, most of the work has been focused on the development of network architecture, particularly on network routing protocols and Medium Access Control (MAC) protocol design. Whereas, relatively less work has been done on the security aspects. The recent history of the Internet and cellular networks has shown that if a given network architecture is not designed with proper security mechanisms from the very beginning, the security vulnerabilities will be exploited by malicious users, and the network might be paralyzed by various types of attacks, for example, the exploitation of WEP security in the 802.11b protocol and the ICMP ping protocol that was basically used for administrative purposes but could result in the 'ping-of-death' due to Smurf attack. Moreover, addressing security issues as an after thought can be expensive and inefficient [24]. Thus, incorporation of security aspects in the networking architecture is a very important requirement for developing solutions for mobile ad hoc networks.

In the context of wired networks, security has been extensively studied and different applicable solutions are already in use. However, due to the salient features (e.g. infrastructureless, self-organizing, wireless channels, mobility, limited resources) of mobile ad hoc networks, most of the security solutions for wired networks may not be applicable in this environment. Many new challenges [50] associated with this environment include:

- Wireless channels suffer from poor protection and are more susceptible to various forms of attacks such as passive eavesdropping, active signal interference, and jamming.

- Ad hoc network routing protocols are co-operative in nature and rely on an implicit trust relationship among participating nodes to route packets. This co-operative nature makes it much easier for data tampering, impersonation, and denial of service (DoS) attacks.

- The lack of a fixed infrastructure and a central concentration point makes some conventional security mechanisms difficult to apply. For example, it makes it difficult for an intrusion detection system to collect audit data (that is usually available at some central points, such as routers or gateways in the case of wired networks), and also impedes the deployment of widespread asymmetric cryptography due to the lack of a PKI (Public Key Infrastructure), where a centralized authority is needed.

- Mobile devices usually have limited memory, slow processing, low battery power, as well as limited radio transmission bandwidth, all of which limit the practical deployment of computationally intensive or more comprehensive security schemes in mobile ad hoc environments.

- Continuous and unpredictable mobility in ad hoc networks makes the detection of malicious behavior (anomaly) and normal behavior (normality) difficult.

Next, we discuss the security aspects in more detail and describe some existing solutions.

2.2 Security challenges

In order to assess network security, to evaluate various network mechanisms, and to choose security products or policies, the following three aspects are usually considered: security attacks, security mechanisms, and security services [49]. The salient features of ad hoc

networks pose new challenges in each of these security aspects, which are discussed below.

2.2.1 Security attacks

A security attack is any action that compromises the security of information illegally, disrupts network proceedings, or degrades network performance. The attacks can be classified into two categories: passive attacks and active attacks [50]. A passive attack obtains information without proper authorization, for example, through eavesdropping or traffic analysis. An active attack involves some type of information interruption, modification, or fabrication. Some examples of active attacks include masquerading (impersonating), replaying, modification of messages, and denial of service (DoS).

In general, all the attacks to which wired networks are vulnerable are also possible in mobile ad hoc networks, however, they also have some unique vulnerabilities that an attacker can exploit. The first vulnerability is due to the wireless channel. The wireless channel is broadcast in nature so it is more vulnerable to various forms of attacks such as passive eavesdropping, active signal interference and jamming. Messages transmitted over the air can be intercepted by the attacker, and faked messages can be injected into the network from anywhere without having physical access to the network components. Also, traffic analysis can be launched by adversaries to identify communicating parties and possibly their functionalities. For example, in a network without precautions against traffic analysis, an adversary may monitor the traffic activities, the nodes with heavy traffic might be the commanding nodes or critical nodes for network connectivity. With this knowledge, the adversaries may be able to take out the important nodes to destabilize the network.

Secondly, nodes in ad hoc networks are vulnerable to physical attacks since the nodes usually reside in an open and potentially hostile environment rather than a physically protected place. In a battlefield scenario, the node itself may be captured or compromised, in such a case, other attacks such as impersonation, message tampering, changing traffic flow, and denial of service (DoS) are possible.

Another vulnerability is due to the co-operative nature and the implicit trust relationship among the participating nodes to relay packets. Due to limited bandwidth, many ad hoc routing protocols are on-demand, which makes them different from the conventional wired network routing protocols. The cooperative nature makes both routing protocols and media access protocols more vulnerable to data tampering, impersonation and denial of service attacks. Just as the wireless medium makes it easier for an attacker to inject false information into the network, the unpredictable and frequent topological changes make it difficult to distinguish between faked routing information generated by malicious nodes and out-of-date routing information caused by topological changes.

2.2.2 Security mechanism

A security mechanism is the mechanism that is designed to provide one or more security services by detecting, preventing, or recovering from one or more security attacks [50]. Different security mechanisms have been proposed, widely used, and proved effective in wired networks, but no single mechanism provides all the services required in a network. Due to certain characteristics of ad hoc networks, some security mechanisms are not applicable to this environment.

First of all, because of a lack of fixed infrastructure and central administration in a mobile ad hoc network, some conventional security mechanisms based on centralized online servers are inapplicable. For example, the conventional public key cryptography scheme based on a centralized Certification Authority (CA), and intrusion detection systems requiring a central concentration point to collect audit data cannot be implemented in ad hoc networks.

Secondly, since nodes in these networks are moving continuously in an unpredictable way, any solutions with a static configuration cannot be implemented. The mobility causes frequent topological changes, so it can be difficult to distinguish between faked routing information and stale routing information. Also, due to mobility, there are more

chances of packet loss and connectivity is uncertain; therefore proper key management is not guaranteed.

Finally, mobile devices are usually resource constrained with limited memory, bandwidth, computational, transmission and battery power. This severely restricts the use of computational intensive, yet more effective, security schemes (e.g. public key cryptography) in mobile ad hoc networks.

2.2.3 Security services

A security service is a service that enhances the security of the network and the information transferred over the network [50]. Security services can be categorized into: confidentiality, authentication, integrity, non-repudiation, access control, and availability [49]. These services are intended to counter one or more attacks, and make use of one or more security mechanism to achieve their goals. In the following we discuss these services in the mobile network environment.

Availability

Availability requires network services to be available to authorized parties whenever needed. Any loss or reduction in availability can result from a variety of denial of service (DoS) attack. In a mobile ad hoc network, an adversary could jam the radio frequencies to interfere with signals on physical channels. It can interact with a node in a legitimate way in order to consume its battery power. It can disrupt routing to cripple the network or can bring down higher level services, such as key management services.

Authentication

Authentication ensures that the origin and the destination of a message is correctly identified, with an assurance that the identities between two communicating parties are not falsified [50]. Without authentication, an adversary could masquerade as a legitimate

node, interfere with other nodes' communication, or gain unauthorized transmission and reception. Mobile devices are susceptible to loss, theft, and capture, thus frequent re-authentication is required. The absence of an online server poses a fundamental problem as the usual authentication mechanisms involve a centralized system entity. This environment requires multiple trusted entities within the network with sufficient availability to provide authentication service to the nodes.

Confidentiality

Confidentiality is the protection of transmitted data from passive attacks, such as eavesdropping. Sensitive information, such as tactical military information or strategic information, requires confidentiality. The other aspect of confidentiality is the protection of traffic flow from analysis. Routing information needs to remain confidential in certain cases, because the source and destination, frequency, length, or other characteristics of the traffic might be helpful for enemies in identifying and locating their targets in a battlefield, or inferring certain tactical information.

Integrity

Integrity ensures that the transmitted information is not illegally modified, for example, by changing, deleting, creating, delaying or replaying of transmitted messages. Certain modifications could be caused by either benign failures, such as radio propagation impairments, or by malicious attacks. Adversaries in mobile ad hoc networks could change routing information in order to disrupt network functioning and to alter network traffic flow.

Non-repudiation

A non-repudiation service guarantees that neither the sender nor the receiver of a message is able to deny the transmission. Non-repudiation helps to detect and punish compromised or misbehaving nodes.

Access control

Access control is the ability to limit and control access to devices and applications via communication links [50]. Each entity attempting to gain access must first be authenticated.

2.3 Available security solutions

This section consists of some proposed solutions to different challenges and attacks.

2.3.1 Defending against physical attacks

As mobile devices are small, light and easy to carry, they are susceptible to loss and theft. In a battle field scenario, they are at risk of being hijacked or captured. Thus, it is necessary to protect the physical security of the mobile devices. The conventional solution to the physical attack is to implement a security module that is tamper-resistant. An example is the use of smart cards, which contain a microprocessor and all necessary cryptographic information. However, when all the information is stored in the smart card, there is still the problem, for example, that the device may be stolen with the smart card in it.

An additional protection scheme to guard against capture is the use of user identification and authentication. Some well known techniques include PINs (personal identification numbers), passphrases, and biometrics. However, frequent re-authentication is somewhat troublesome, and discourages users from activating the security mechanisms which provide authentication at login time.

2.3.2 Enforcing confidentiality

Since the wireless channel in mobile ad hoc networks is broadcast in nature, it suffers from poor protection and is particularly vulnerable to passive eavesdropping attacks.

This vulnerability is not specific to mobile ad hoc networks, but is common to all wireless networks. Confidentiality in this environment may consist of two aspects: one is to protect the identity of nodes (either users' identities or the communication entities' functionalities), and the other is to protect the transmitted messages from disclosure. In considering the first aspect, we discuss hiding nodes' transmissions and methods to avoid traffic analysis. Whereas for the second aspect, we discuss methods for securing the communication path or the messages themselves.

Hiding the nodes' transmission

One way to protect the identities of communicating entities, is to effectively conceal their communications. This can be achieved through solutions related to the physical layer, such as spread spectrum technologies, which either spread the energy in time and/or frequency in a random fashion to make signal capture difficult or spread the energy to a wider spectrum so that transmission power is hidden behind the noise level. Directional antennas can also be used due to the fact that the communication techniques can be designed to spread the signal energy in space [50].

Preventing traffic analysis

Traffic analysis is basically the attempt to discover the pattern of traffic between parties. There are two approaches to prevent traffic analysis through encryption: end-to-end encryption and link encryption. End-to-end encryption is performed at or above the transport layer in two end systems. When the packets are transmitted over the network, the payload or message content is all encrypted, except for the header. So the eavesdropper is able to get end-to-end traffic flow pattern information such as source and destination, frequency of packets exchanged, duration, etc. from the header. This information can help the adversary to locate the target or infer the activity or intention of the communicating parties. The link encryption is performed at the data link layer. Since, multiple end-to-end flows may be multiplexed at each link, an eavesdropper would not be able to distinguish the traffic pattern of end-to-end flows, however, the link flow traffic is still observable. The link encryption is a better security mechanism for hiding end-to-end traffic

flow than end-to-end encryption.

Other solutions to prevent traffic analysis include, creating security clouds in such a way that each node under a security cloud is identical in terms of traffic generation. Use of the dynamic mix method (DMM) [42] can also hide source and destination information. It achieves the anonymity of message delivery via a cryptographic method and the *mix* nodes [16] in the network. Due to dynamic topological changes, network performance degrades with the conventional solution of mix nodes. Whereas the DMM technique improves network performance by allowing the communicating nodes to choose mix nodes dynamically at run time.

Securing the communications path

In order to secure communications paths, one common approach is to encrypt all messages exchanged between communication entities, either encrypted from point-to-point or end-to-end. However, due to resource constraints such as bandwidth, memory, computational and battery power, the full version of security schemes (e.g. public key cryptography) used in wired networks may not be effective in mobile networks. Security solutions with minimum resource requirements are required for this environment.

Another approach for enhancing confidentiality is using multipath routing. The idea is to utilize the salient features of mobile networks, such as the mobility of the network architecture. A messenger who carries the full message from one place to another across hostile ground may reveal the message easily if he/she is captured. Whereas the message will not be fully recovered if multiple messengers are deployed to carry only partial information and go through different routes across the hostile ground.

2.3.3 Authentication

The process of verification by a node to ensure that the other node with which it communicates, is what it claims to be, is called authentication.

One method for authentication is the challenge-response protocol. If a node shares a secret with another node, the node can issue a challenge message and request the correct answer. Upon receiving the correct answer, the other node is trusted to be the one sharing that secret. If both parties do not share a secret, a trusted third party (Certificate Authority (CA)) can be used for the verification with the understanding that both parties share secrets with the CA. Then the challenge process will be carried out between both parties through the CA. In the traditional Internet, centralized CAs in the fixed infrastructure exist, but in mobile ad hoc networks such centralized service is not possible. Therefore, several distributed models have been developed.

In the distributed public key trust model in [30], a selected set of nodes is used as servers or CAs that collaboratively manage the public keys. Whenever a node needs to have its public key signed, it has to contact a subset of servers to gain the certificate of its public key, which is used in the authentication process when communicating with other nodes. In another solution in [24], the authors suggest a self-organized public key infrastructure based on a chain of trust, instead of pre-selecting a set of nodes as the CAs as in [30]. Whenever a node wants to communicate to another node, the trust repositories of both parties will be merged and a search for a trust chain is initiated. Any commonly trusted entity will be the basis of developing trust among the communicating nodes. In [8], a light-weight authentication model is proposed. This trust model is based on human behavior. If a node A wants to authenticate node B. Node A would ask about a secret or recent transaction by node B. If that does not work, A will start to solicit A's trusted nodes for recommendations, or asks B to provide a list of references for verification. If A's trusted friend node says yes, or a reference from B can be authenticated by A and tells A that B is trusted, then A can trust B.

2.3.4 Key management

Key management is one of the most critical and complicated issues in the security of mobile ad hoc networks, because the applicability of many other services, such as confidentiality and authentication, relies on effective and efficient key management [50].

In the construction of secure information distribution systems, cryptography has been widely used. In addition to other factors, the strength of cryptographic systems depend on proper key management. In symmetric cryptography, if an attacker compromises the symmetric key of a group of users, then all encrypted messages for that group will be compromised. Whereas in asymmetric cryptography, although compromise of a private key of a user does not reveal messages encrypted for other users in the group, it is nevertheless computationally expensive. In practice, symmetric cryptography is widely used for bulk data encryption while asymmetric cryptography is used to distribute cryptographic keys as its performance is inadequate for encryption of data. The management challenge is to make public keys accessible to all concerned parties without the potential abuse of the public key distribution system.

One approach to providing public key management service in wired or wireless cellular networks is the deployment of a Public Key Infrastructure (PKI). The success of PKI depends on the availability and security of a Certification Authority (CA). Thus, a PKI requires a central control point, which everybody trusts. The difficulty in applying a PKI in a mobile ad hoc network is that such a central control point is not feasible. Even if it is deployed, it cannot be well protected and would become the most vulnerable point in the system. There are two major research directions for solving this problem.

- One is to retain the certificate authority concept, but distribute its functionality into multiple servers or trusted nodes. In this way, both the availability and the security of the CA can be improved.

- Another approach is to discard the centralized CA, and instead, create a completely distributed and self-organized key management system.

Distributed and cooperative certificate authority

A distributed and cooperative CA model was proposed by Zhou and Haas in [30]. In this model, the distribution of trust is achieved by using threshold cryptography. An $(n, t+1)$ threshold cryptography scheme allows n parties to share the ability to perform a cryptographic operation (of signing public key certificates), such that any $t+1$ or more parties can perform this operation jointly. It is infeasible for t or fewer parties to do so, even in collusion. With an $(n, t+1)$ configuration where $(n \geq 3t+1)$, there are n special nodes called *servers*, which collectively perform the functions of a CA. The system private key K is divided into n shares $(s_1, s_2, \cdots, s_n)$, one share is given to each server. Each server has its corresponding private/public key pair and stores the public keys of all the nodes in the network. In particular, each server knows the public keys of all other servers. For the service to sign a certificate, each server generates a partial signature for the certificate using its private key share and submits the partial signature to a combiner. With $t+1$ correct partial signatures, the combiner is able to compute the signature for the certificate. In order to protect against an attack in which an adversary, after compromising a server, attempts to compromise others, the authors proposed to use share refreshing. Share refreshing allows the servers to compute new shares from the old shares in collaboration without disclosing the system's private key to any server.

Self-organized public key management

Similar to PGP (Pretty Good Privacy), where there is no centralized key certification authority, instead every user generates and distributes his/her own public key and users then sign each other's public keys creating an interconnected community of PGP users, an alternative key management system was proposed by Hubaux *et al.* in [24]. In this system, each user maintains a local certificate repository that contains a limited number of certificates selected by the user according to some algorithm. When user u wants to obtain or verify the public key of user v, user u will merge his/her local certificate repository with that of user v, and will try to find an appropriate certificate chain from u to v in the merged repository. The authors use a directed graph called a trust graph to represent the trust relationship between users.

The solution in [24] is a fully distributed, and scalable approach to key management. However, it only provides probabilistic guarantees of finding trust chain. In addition, this approach assumes that trust is transitive, which is often not the case in practice. In order to alleviate this problem, the authors proposed using multiple certificate paths and using authentication. Similar to PGP, the weakest link in this whole system is key revocation [50] i.e. if any user is revoked, it can disrupt the whole chain of trust.

2.3.5 Handling node misbehavior

In a mobile ad hoc network, all basic functions such as routing and packet forwarding are collaboratively carried out by the participating nodes. The effectiveness of the network relies heavily on mutual trust and mutual collaboration in sharing network resources. It is expected that all participants follow the rules according to the network design objectives. However, there might exist some selfish nodes that do not want to provide services to other nodes for reasons such as: to save their own battery energy, or to grasp more bandwidth or to demand less delay for their own packets. Such misbehavior can cause significant network performance degradation.

Routing and packet forwarding are two closely related functions in the network layer, but the correctness of the routing information does not guarantee the correct forwarding of a packet. Two types of solutions have been proposed to deal with reluctant or even erroneous packet forwarding problems. The first type is reactive in nature; the misbehaving nodes are detected and corresponding reactions to these nodes are carried out. The other type is to create an incentive mechanism to encourage the cooperation of the nodes [50].

In [44], a technique is proposed to tackle the problem of nodes that do not forward packets; a watchdog is used to identify misbehaving nodes and a pathrater is designed to help routing protocols avoid such nodes. The watchdog's mechanism is based on the

promiscuous mode of radio interface: the receiver of one node could listen to the transmission of any of its neighbours, regardless of the intended destination of that transmission [50]. Thus, when a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet by overhearing the next node's transmission. If the next node does not forward the packets as expected, then it is misbehaving. The pathrater then uses this knowledge to rate the nodes and chooses network paths to avoid the use of misbehaving nodes. Since this scheme is based on the idea of avoiding the problem rather than facing it, if there are too many such misbehaving nodes the pathrater may not be able to find a feasible path, which will degrade the network performance severely.

Another protocol in [41] uses a similar technique to that in [44]. It consists of four major components: the Monitor, the Reputation System, the Path Manager, and the Trust Manager. It improves the detection mechanism by letting nodes learn not only from their own experience, but also by exchanging experiences with their neighbours. Instead of avoiding the misbehaving nodes, it isolates the detected selfish nodes so that misbehavior is punished and cooperation is rewarded.

In an incentive based solution in [29], two models based on virtual currency called "nuglets" are proposed: the Packet Purse model, and the Packet Trade model. In the packet purse model, when a node originates a packet, it puts an estimated amount of nuglets in the packet purse. Each node forwarding the packet would then take a certain number of nuglets from that purse. If the packet runs out of the nuglets, it is dropped. In the later version of this scheme, the packet purse is removed, and a nuglet counter is used. Whenever a packet is originated, the number of nuglets is reduced from the nuglet counter of the source, and the counter is increased when a node forwards the packet. This approach requires tamper-resistant security hardware, and also requires estimation of the cost from source to destination.

2.3.6 Intrusion detection

Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user. In general, there are two approaches for detecting an intrusion: misuse (or rule-based) detection and statistical anomaly detection [50].

The misuse detection system attempts to define improper behavior based on the patterns of well-known attacks. It can accurately and efficiently detect known attacks but it lacks the ability to detect unknown attacks. The statistical anomaly detection system attempts to define normal or expected behavior. It involves the collection of data relating to the behavior of legitimate users over a period of time, and by applying statistical tests it analyzes whether an observed behavior deviates from legitimate user behavior.

The intrusion detection systems (IDSs) designed for wired networks cannot function well in mobile ad hoc networks. Because the IDS relies on real time traffic analysis of traces collected at switches, routers, or gateways, whereas in mobile networks there are no such traffic concentration points from which one can collect audit data for the entire network. Also, mobility in these networks complicates detection because it is harder to differentiate between anomalous and normal behavior. Thus, more sophisticated solutions are required for intrusion detection that can give reliable alternate sources for collecting audit data.

2.3.7 Securing routing protocols

For mobile ad hoc networks, many routing protocols have been proposed, which can be divided into two categories: table-driven (proactive) routing protocols and on-demand (reactive) routing protocols.

Table-driven routing protocols attempt to maintain consistent, up-to-date route information from each node to every other node, regardless of the need for such routes. They respond to changes in topology by propagating updates throughout the network, and thus create lots of communication overhead.

On-demand or source routing protocols attempt to discover a route to a destination only when it has a packet to send to that destination. Discovered routes are maintained by a route maintenance procedure until either the destination becomes inaccessible along the path from the source or the route is no longer desired.

Vulnerability analysis of routing protocols

Attacks on a routing protocol can be divided into two classes: passive attacks and active attacks. *Passive attacks* include eavesdropping and traffic analysis. They are mainly a threat to message confidentiality, and do not affect the proper functioning of the routing protocols. Passive attacks on routing protocols can be avoided by protecting the data traffic. *Active attacks* can be further classified into external and internal attacks. External attacks are from outsiders; such attacks are limited when encryption and source authentication are utilized. An example of an external attack is the wormwhole attack. Internal attacks, due to internal nodes (either malicious authorized nodes or compromised nodes) are more severe because an internal node has all the necessary credentials for authentication, information about critical nodes in the network, and so on. A single node or multiple nodes could launch attacks individually without collusion, or multiple nodes could also launch attacks as a team. Following are some major attacks and brief descriptions of possible preventions.

Modification of routing information: In order to disrupt the routing function, a general attack is to modify the routing information. For example, for a table driven protocol, a malicious node could send out false routing updates, or for an on-demand protocol, a malicious node could alter the information contained in the route request or route reply in a route discovery reply. The altered routing information could cause legitimate traffic to be directed to black hole[1], or a routing loop is formed, or even a network partition may be caused. Any information field in a routing message could be exploited by an attacker. For example, the *destination-sequence-number* is widely used in mobile ad hoc network

[1]a black hole is where all the packets are dropped except routing packets

routing protocols to indicate the freshness of the routes. A malicious node could simply modify it to make other routing information invalid while making itself the freshest route, or it could modify the hop counts to claim the shortest path to any destination passing through it. In protocols such as DSR (Dynamic Source Routing), any intermediate node could simply modify the replied routes, or a malicious node could illegally modify its IP and/or MAC address to impersonate another node.

This type of modification attack could be carried out by either an external attacker or an internal attacker. Such attacks can be prevented by source authentication and message integrity services such as a Message Authentication Code (MAC).

Fabrication of routing information: Another form of attack is to fabricate false topological information. A malicious node can fabricate routing updates or route error messages to claim the inaccessibility of another node, or change network traffic flow. Source authentication can limit this type of attack to the extent that the malicious node could only claim the inaccessibility to its own neighbours. However, non-repudiation protection can be applied to facilitate the detection of such attackers.

Replay attack: A replay attack captures a data unit passively and then retransmits it to produce an unauthorized effect. One example of such an attack in mobile ad hoc networks is the wormhole. In a wormhole attack, the attacker collects packets, particularly routing control packets, at one location in the network, tunnels them to another location, and then retransmits them into the network. This could be either an external attack or an internal attack. This attack is dangerous because the source may fail to find routes or find routes that actually do not exist. In an individual wormhole attack, the node M in between nodes S and T can simply replay routing requests and reply messages without showing itself. Thus, an actually non-existing path $S-T$ would be replied to S. Similarly, two or more malicious nodes can launch this attack collaboratively, for example, node M_1 and M_2 can collude to give impression of shorter path than other normal multihop routes by providing S and T with the route $S-M_1-M_2-T$ while there may exist multiple nodes in between M_1 and M_2 (with actual path being, for example, $S-M_1-x-y-z-M_2-T$),

and thus could attempt to change traffic flow.

The external wormhole attack can be partially prevented at the physical layer, for example, by using a secret modulation method, RF watermarking, or tamper-resistant hardware [50]. Detection of such an attack is possible by using some unalterable and independent physical metric such as time delay or geographical location.

Denial of service (DoS): An attacker could launch a DoS attack by excessively consuming network resources. For example, a malicious node could initiate excessively unnecessary route discovery processes or it could inject extra packets into network with the sole purpose of wasting other nodes' energy when processing and forwarding packets. An attack is considered a denial of service attack if the ratio between the total work performed by nodes in the network and the work performed by the attacker is on the order of the number of nodes in the network. For example, a single packet sent by the attacker results in a packet flood throughout the network. The DoS attack can be limited by preventing the attacker from inserting routing loops, or enforcing a maximum route length a packet can traverse.

Protection of routing protocols

In general, a secure routing protocol should implement some kind of authentication and integrity schemes so that the correctness of the routing information, particularly the node identity, the *destination-sequence-number*, and the cost metric, can be protected. Following are the methods incorporated with different proposed protocols.

Hop-by-hop authentication: For distance vector routing protocols such as DSDV and AODV, due to their operational features, the correctness of the routing information has to be provided on a hop-by-hop basis. The correct accumulation of the routing metric has to be guaranteed, i.e. an internal malicious node should not be able to reduce the routing metric from itself to any other destination. In SEED [52], the authors suggest the Message Authentication Code (MAC) for neighbour authentication and to protect fields

such as source identity and destination sequence number, with each routing update. They proposed a one-way hash chain to protect the metric field. The one-way hash chain is a series of data generated from a one-way hash function, which is easy to compute in one way while infeasible to do in the reverse. In [28], the authors use public key cryptography (digital signature) at the hop-by-hop level to guarantee message authentication, integrity and non-repudiation.

End-to-end authentication: For source routing protocols such as DSR, authentication can be done on an end-to-end basis because the end nodes have the knowledge of the complete route. However, the integrity of the routes (the complete and correct node list) needs to be carefully protected in this case. Message authentication codes (MACs) have also been suggested for message authentication and integrity protection in end-to-end authentication, such as in Ariadne [52].

Reactive approach: The above mentioned solutions are proactive and preventive methods for securing routing protocols. The objective is to secure the correctness of the exchanged routing information as well as the routing operation [50]. One motivation for using reactive mechanism is to protect against attacks that are difficult to prevent. One solution (based on the reactive approach) to the wormhole attack comes from the use of additional timing and/or location information so that a receiver can determine if the packet has travelled a route that is not realistic for the specific network technology used. A reactive approach can monitor the behaviors of both the routing function and the forwarding function, so another motivation for a reactive approach is to provide unified network layer protection.

2.4 Conclusion

In light of the above discussion, we conclude that almost all the attacks possible in wired networks are also possible in wireless ad hoc networks, and many of those become more challenging due to the nature of the typical ad hoc network. In this thesis, we mainly focus on the security services of authentication, non-repudiation, confidentiality, and integrity.

We also address key management issue, but it is not completely distributive. For identity based key generation and distribution, we describe a method for using multiple Private Key Generation authorities (PKGs) instead of centralized key generation authority. The use of identity based keys makes key management simple as compared to conventional key schemes. In addition, we consider the issue of reducing communication, computational, and memory overhead. This overhead reduction is accomplished through the use of non-interactive computation of pairwise keys and unique broadcast keys, use of symmetric cryptography in pairwise communication, use of group keys in group communication, and having the flexibility of discarding identities or keys that are not in frequent use thereby reducing memory requirements.

Chapter 3

Identity-Based Cryptography (IBC)

The solutions that we propose in this thesis for wireless ad hoc networks mostly use Identity-Based Cryptography (IBC). In this chapter, we will discuss identity-based cryptography, its working procedure, the algorithms to implement it, and give the basic terminologies required to explain the technical details.

3.1 Introduction

Cryptography has been widely used to achieve security services, such as confidentiality, authentication, integrity, and access control. These services can be used to defend against different kinds of attacks, as described in Chapter 1. Such services can either be achieved with symmetric key cryptosystems (e.g. AES, DES, RC5) or with public key cryptosystems (e.g. RSA, ElGamal). Both systems have some merits and demerits, and are used in appropriate contexts. For example, public key cryptosystems are normally used for secure key exchange, whereas symmetric cryptosystems are used for secure bulk data exchange.

In general, public key cryptosystems are based on a hard problem. Some well known cryptographic systems, such as the ElGamal cryptosystem and the Diffie-Hellman key exchange protocol, depend on the Discrete Log Problem (DLP) described in Appendix A. However, DLP is easy to solve in additive groups over a finite field, and there are methods to solve DLP in multiplicative groups in sub-exponential time. Hence, such

systems should be used with large fields of multiplicative groups, and thereby require large key size and expensive computation. This situation led V. Miller and N. Koblitz to propose a technique for replacing the finite field group with the group of rational points on an elliptic curve. It turned out that the DLP in additive elliptic curve groups is one degree of magnitude harder to solve than the corresponding problem in the multiplicative group of a finite field of similar size [22]. In light of the research on this comparison, Blake *et al.* in [22] show that, with the current algorithmic knowledge, the key size in an elliptic curve cryptosystem grows slightly faster than the cube-root of the corresponding conventional key size, of similar cryptographic strength. An example of this comparison shows that, for the conventional key sizes of 1024 and 4096 bits (common values for RSA), the equivalent key size (providing similar security) with elliptic curves are 173 and 313 bits respectively. Thus, the authors in [22] conclude that, in practice, shorter key lengths can translate to faster implementations, less power consumption, less memory consumption and less silicon area requirement. These results were our motivation to use elliptic curve cryptography to find security solutions for wireless ad hoc networks that have the same requirements. Moreover, we identified additional features of identity-based cryptography (which is also an elliptic curve cryptography) that could yield even better results for this environment. In the following, we discuss identity-based cryptography in detail and give some examples of its applications.

3.2 Identity-Based Encryption

Identity-based encryption is a cryptosystem in which both public and private keys are based on the identities of the users. The idea of Identity-Based Cryptosystem was first proposed by Shamir [7] to simplify the conventional Public Key Cryptosystem (PKCS), and make key management easier. Since then, different schemes were proposed such as [45, 21, 53], but some were based on unrealistic assumptions and some required tamper resistant hardware. The first practical IBE was proposed in [15], which is based on Elliptic Curve Cryptography and the pairing of points. Since in this scheme the identity of the user is used as the public key, key management is simplified and does not require

certificates for implementing user-key binding. This minimizes the overhead of the conventional PKCS and becomes an attractive scheme for mobile and resource constrained devices.

Problems in the conventional PKCS: The purpose of introducing identity-based encryption was to simplify the conventional Public Key Cryptosystem. The problem in public key cryptosystems is that they create overhead of certificate management for both the Certification Authority (CA) and the users.

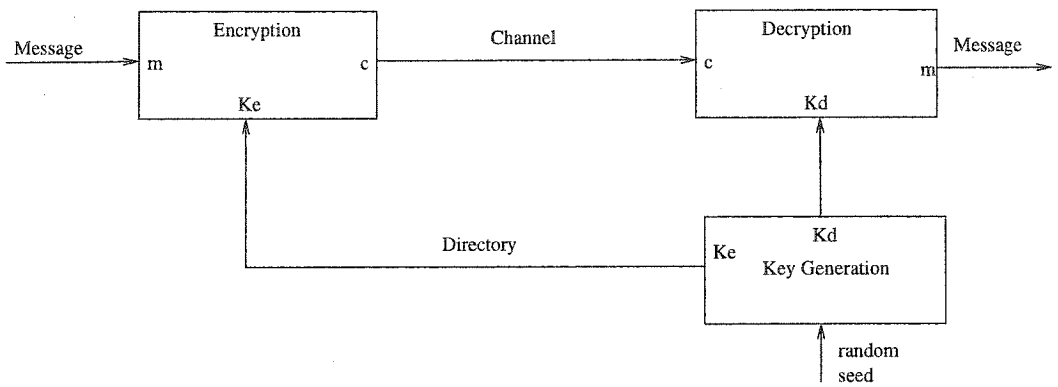
Whenever the private key of a user is expired, the owner has to get a new private key approved by the CA and other users have to contact the CA to get the new public key certificate for the corresponding new private key. Such off and on updates, requiring contacting the CA, are inconvenient from the user's point of view.

The users have to keep the certificates of other users even if they are not in frequent use. This causes memory consumption and is inappropriate for resource constrained devices. If such certificates are discarded in order to save memory, users will have to contact the CA again to get them when required. However, access to the CA may not be possible all the time, which may cause further delays.

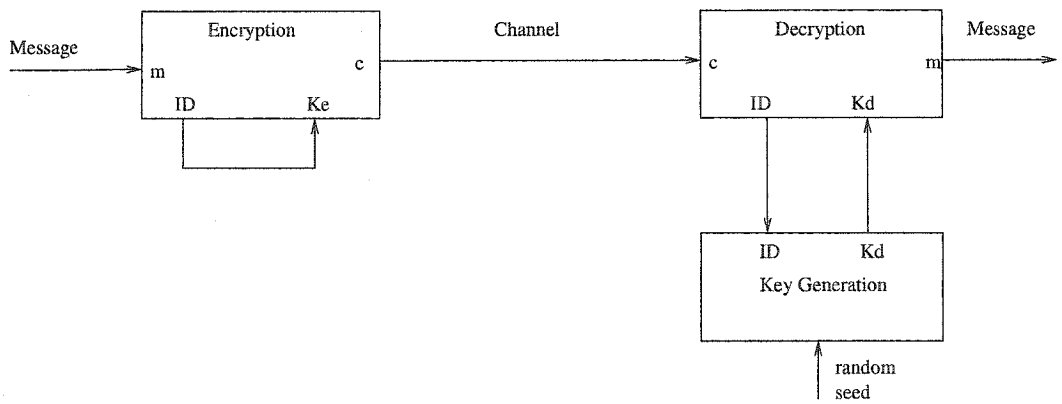
The revocation or re-issue of one private key triggers lots of communication in the network. Since a user in the organization may have contacts with several other users, say n, if one private key is revoked or reissued, then other users will also have to get the new public key certificate for that user. Thus, the renewal of n private keys in the organization can require up to n^2 public key certificates to be redistributed. Such renewal periods therefore cannot be very frequent, perhaps not even daily.

The CA, has to maintain the certificate list it has issued and to implement the revocation process efficiently. However, swift revocation is a difficult problem and is not solved in identity based cryptography. Instead, solutions to this problem depend more on security mechanisms in use than on a cryptographic service.

Bellow is a block diagram of identity-based cryptosystem compared to conventional public key cryptosystems.



(a) Conventional Public Key Cryptosystem (PKCS)



(b) Identity–Based Cryptosystem (IBCS)

Figure 3.1: Block diagram of the conventional PKCS and the IBCS

Applications and advantages: In IBE, unlike conventional PKCSs, the sender can compute the public key of the receiver even before the owner has received his/her private key. This is possible because the public and private keys are not generated by users themselves; instead the Private Key Generating (PKG) authority does that job. When

the user receives an encrypted message, he/she can then contact the PKG and obtain the private key to decrypt the message after proving his/her identity in the same way as with a Certification Authority (CA). Following are some useful applications of IBE as given in [15].

- **Revocation of keys:** Revocating a key or setting an expiration period is simple. It is done by adding date field to the identity, for example, "bob@company.com || current-date". To illustrate the usefulness of this approach, consider an organization that issues private keys to its employees on daily basis. When an employee leaves the organization, the next day the private key would automatically expire and would no longer be useful to decrypt the emails. In addition, even though the private keys are issued on daily basis, there is no need to issue corresponding public keys to other users who might be using that person's public key. This is because, other users can compute the corresponding public key by themselves by changing the date argument. It should be noted that, typically, changing the date argument would change the point for ID on the curve but would not make the private key more secure, because anyone can know the current date and any additional arguments, which are public. However, it is useful for the purpose of key expiration as discussed before.

- **A user as the PKG authority:** In IBE, a user can also become his own key generating authority, instead of getting the private key from a third party PKG. In the later case, the master key remains with the PKG who can then decrypt any messages belonging to any of its users. With the approach of being a self-PKG, one can generate as many keys as needed. However, one will need a certificate from the CA for the purpose of authenticating his identity to other users. This is necessary to make other users trust that the parameters they are receiving from person 'A' have really come from that person, otherwise some one else can pretend to be that person. Thus, one can generate as many keys as required, for example, for the purpose of delegation of duties and can revoke such keys whenever needed.

For example, Bob has several assistants, say, one for Marketing, one for technical

matters and one for purchasing. Bob can create the respective private keys for each of his assistants based on his ID just by adding the subject of the duty of the assistant. Whereas other users having Bob's public key can send email to Bob's address with the required subject matter related to the duty of the assistant and can get a response from the appropriate person. No assistant would be able to decrypt the email whose subject differed from his/her duty. In short, there can be multiple private keys and one master public key i.e. one part of the public key will remain the same while the remainder, based on the duty, will be changed and can easily be computed by knowing the ID and subject matter. Hence, there is no need to send different public keys for all private keys that are based on one master key. This example can be illustrated in the following figure.

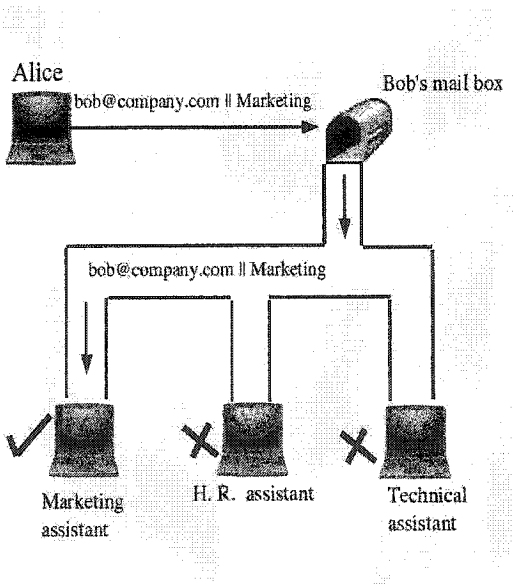


Figure 3.2: Delegation of duties through IBE

Key escrow problem: One drawback with the IBE system is that it suffers from the key escrow problem. A key escrow based cryptosystem allows a trusted authority (by providing it sufficient information) to decrypt the messages of any of its users, and thus affects the privacy of users. Since the private keys of users are computed by the PKG

itself, it has access to all the messages encrypted for such private keys. Thus, in IBE systems, the users need a trustworthy PKG; however, it is possible to use multiple PKGs through threshold cryptography instead of using a single PKG. This can enhance the privacy of users, and also the security of the master key.

Essential elements of the IBE system:

There are four basic algorithms of the IBE system:

Setup, Extract, Encrypt, and Decrypt

The *Setup* algorithm generates system parameters, such as elliptic curve field parameters, a random generator point, hash functions etc. *Extract* is used to generate private keys for users. The *Encrypt* algorithm is used for encryption and *Decrypt* for decryption.

Encryption and decryption are based on bilinear maps between two groups of points on a curve, which is defined as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. This can be achieved by pairing of points, e.g. Weil Pairing or Tate Pairing. Such pairings have the Bilinear property; a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if:

$$e(aP, bQ) = e(P, Q)^{ab}$$

The strength of the IBE system is based on the Bilinear Diffie-Hellman Problem (BDHP) and the Elliptic Curve Discrete Log Problem (ECDLP). The hashing operation is assumed to be in the random oracle model. The elliptic curve used is a super singular curve, e.g. $y^2 = x^3 + 1$. The following is a further explanation of the elements above.

Supersingular curve: An elliptic curve is said to be supersingular if it has an endomorphism to a ring of points which is non-commutative. The endomorphism is a surjective (onto) mapping from one group of points of the curve to another group of points on the same curve. There are other criterion as well, for verifying that a curve is supersingular,

but in the context of IBE cryptosystems this is the most relevant feature as the IBE encryption process is based on the non-commutative endomorphism of points of the supersingular curve. The IBE encryption operation is described in the following.

The Bilinear Diffie-Hellman (BDH) Problem: The BDH problem is a variant of the Computational Diffie-Hellman (CDH) problem and the Decisional Diffie-Hellman (DDH) problem. The simplest among these is the Decisional Diffie-Helman (DDH) problem in group $\mathbb{G}_1$, which requires one to distinguish between the distributions $< P, aP, bP, abP >$ and $< P, aP, bP, cP >$, where a, b, c are random in $\mathbb{Z}_q^*$ and P is random in $\mathbb{G}_1^*$. Whereas CDH in $\mathbb{G}_1$ is harder than DDH and it requires one to find abP, given the random $< P, aP, bP >$.

The BDHP states that, if $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups of prime order q and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map, and if P is the generator of $\mathbb{G}_1$, then given $< P, aP, bP, cP >$ for some a, b, $c \in \mathbb{Z}_q^*$, compute $W = e(P, P)^{abc} \in \mathbb{G}_2$.

The Random Oracle Model: The random oracle model is a mathematical model of an ideal hash function. In this model, a hash function $h : X \rightarrow Y$ is chosen randomly from $F^{X,Y}$ (the set of all functions from X to Y), and we are only permitted oracle access to the function h. This means that we are not given a formula or an algorithm to compute values of the function h. Therefore, the only way to compute a value $h(x)$ is to query the oracle. This can be thought of as looking up the value $h(x)$ in a giant book of random numbers such that, for each possible x there is a completely random (but fixed) value $h(x)$.

For IBE systems, the hash functions H_1, H_2, H_3, and H_4 are ideally required to be queried in a random oracle model. In the real world, the possible choices are SHA-1, 128 or 256.

The mapping function: MapToPoint

As described in [15], the MapToPoint function encodes the identity of the user to a

point on an elliptic curve as follows.

Let E be the elliptic curve $y^2 = x^3 + 1$ over $\mathbb{F}_p$, where $p = 2 \ mod \ 3$, and $p = lq - 1$ for some prime $q > 3$. Also q^2 must not divide $p + 1$. Let $\mathbb{G}_1$ be an additive subgroup of points on $E(\mathbb{F}_p)$ of order q. We define the hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{F}_p$, which gives the hash of an input ID of any length and the resulting value is an element in the field $\mathbb{F}_p$. Let $y_0 = H_1(ID_X)$, where ID_X is the identity of any user X. The MapToPoint algorithm works as follows on the input $y_0 \in \mathbb{F}_p$:

(1) Compute $x_0 = (y_0^2 - 1)^{1/3} = (y_0^2 - 1)^{(2p-1)/3} \in \mathbb{F}_p$.

(2) Let $Q = (x_0, \ y_0) \in E(\mathbb{F}_p)$, and set $Q_{id\_X} = lQ \in \mathbb{G}_1$.

(3) Output $\textbf{MapToPoint}(y_0) = Q_{id\_X}$.

The resulting point Q_{id_X} is the identity based public key of the user X.

The IBE algorithms

The following is a description of the four essential algorithms of the IBE system proposed in [15].

(1) Setup

Step 1: Take an input security parameter $k \in Z^*$, which can be used as the number of bits for order q of the groups of points, e.g. $k = 512$ bits or 1024 bits.

Step 2: Run the BDH parameter generator which should generate a k-bit prime q, and find a smallest prime p such that (1) $p = 2 \, mod \, 3$, (2) q divides $p + 1$, but (3) q^2 does not divides $p + 1$. It also generates subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the curve $y^2 = x^3 + 1$ of order q over the field $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ respectively, where $\mathbb{G}_1$ is an additive group of points on the curve $E(\mathbb{F}_p)$ and $\mathbb{G}_2$ is the multiplicative group over $E(\mathbb{F}_{p^2}^*)$. A bilinear map e is defined

as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

Step 3: Choose an arbitrary generator $P \in \mathbb{G}_1$.

Step 4: Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$.

Step 5: Choose the following hash functions:
(1) $H_1 : \{0,1\}^* \rightarrow A \in \{0,1\}^*$, and an encoding function $L : A \rightarrow \mathbb{G}_1^*$.
(2) $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$, for some n.
(3) $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$.
(4) $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$.
The message space is $M = \{0,1\}^n$. The cipher text space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. Thus, the system parameters are:
Params $= < q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, H_1, H_2, H_3, H_4 >$
and the *master−key* is $s \in \mathbb{Z}_q^*$.

(2) Extract

Step 1: For the identity (e.g. email address or any string) compute $Q_{ID} = L(H_1(ID)) \in \mathbb{G}_1^*$.

Step 2: Set the private key $d_{ID} = s.Q_{ID}$.

(3) Encrypt

Step 1: Compute $Q_{ID} = L(H_1(ID)) \in \mathbb{G}_1^*$.

Step 2: Choose a random $\sigma \in \{0,1\}^n$.

Step 3: Compute $r = H_3(\sigma, M)$.

Step 4: Compute the cipher text as $C =< rP, \sigma \oplus H_2(g_{ID}^r), M \oplus H_4(\sigma) >$, where $g_{ID} = e(Q_{ID}, P_{pub}) \in \mathbb{G}_2$.

(4)Decrypt

Let $C =< U, V, W >$.

Step 1: If $U \notin \mathbb{G}_1^*$ reject the cipher text.

Step 2: Compute $V \oplus H_2(e(d_{ID}, U)) = \sigma$.

Since, $e(d_{ID}, U) = e(Q_{ID}, P_{pub})^r = g_{ID}^r$

$$V \oplus H_2(e(d_{ID}, U)) = \{\sigma \oplus H_2(g_{ID}^r)\} \oplus H_2(g_{ID}^r)$$

$$= \sigma$$

Step 3: Compute $W \oplus H_4(\sigma) = M$

Since, $W = M \oplus H_4(\sigma)$

$$W \oplus H_4(\sigma) = \{M \oplus H_4(\sigma)\} \oplus H_4(\sigma)$$

$$= M$$

Step 4: Set $r = H_3(\sigma, M)$. Test that $U = rP$. If not, reject the cipher text. If $U = rP$ then M is the decryption of C.

As the IBE system is based on the pairing of points, we explain a method for computing one such pairing, called the Weil pairing, in the following.

3.2.1 The Weil Pairing

Definition: Weil pairing is the mapping of two elements of any group $\mathbb{G}_1$ to an element of group $\mathbb{G}_2$, i.e

$$e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

Let E be an elliptic curve, and P and Q be the elements of the m-torsion group i.e $P, Q \in E[m]$. If A_P and A_Q are divisors with disjoint support such that:

$$A_P \sim (P) - (\mathcal{O}),\ A_Q \sim (Q) - (\mathcal{O})$$

then there exist rational functions f_P and f_Q whose divisors are mA_P and mA_Q respectively. The Weil pairing of the points P and Q is defined as:

$$e_m(P, Q) = \frac{f_P(A_Q)}{f_Q(A_P)}$$

Applications of the Pairing

Following are some applications of pairings of points as described in [27].

(1) Tripartite Key Exchange:

Let three users A, B and C agree on a prime p and a generator g. Then using the two-party Diffie-Hellman key exchange, the following six pass protocol is used:

$A \rightarrow B, C : g^a \ (mod\, p)$
$B \rightarrow A, C : g^b \ (mod\, p)$
$C \rightarrow A, B : g^c \ (mod\, p)$
$A \rightarrow B, C : g^{ac}, g^{ab} \ (mod\, p)$
$B \rightarrow A, C : g^{bc}, g^{ab} \ (mod\, p)$
$C \rightarrow A, B : g^{bc}, g^{ac} \ (mod\, p)$

At the end each party can compute $g^{abc} \ (mod\, p)$.

By using pairings, this could be achieved in only three passes. First they agree on groups $\mathbb{G}_1, \mathbb{G}_2, P$ and the mapping function e, and then following key exchange will take place:

$A \rightarrow B, C : aP$
$B \rightarrow A, C : bP$
$C \rightarrow A, B : cP$

After that, A can compute $e(bP, cP)^a = e(P, P)^{abc}$, and the key will be computed by B and C in an analogous method.

(2) Identity-based Key Exchange:

Two users can compute their shared secret key without any round of information sharing between them provided they must know each other's identity and have received their private keys from a trusted authority. The *TA* will generate their private keys from a master key as follows:

$$A \leftarrow s.Q_A$$
$$B \leftarrow s.Q_B$$

where s is the master key and $Q_A = H(ID_A)$ and $Q_B = H(ID_B)$. Both A and B will compute their shared key as:

$$e(sQ_A, Q_B) = e(Q_A, Q_B)^s = e(Q_A, sQ_B)$$

(3) ID-PKCS:

Identity-based encryption and identity-based signature schemes are based on pairings, which makes a candidate PKCS based on identities.

(4) The pairings have been used for attacking cryptosystems in the past, and can be used to reduce the discrete log problem in elliptic curves to the finite field.

Properties of the Weil Pairing:

An important property of pairing is the *bilinearity* property. According to that:

$$e(2P, P) = e(P + P, P) = e(P, P).e(P, P) = e(P, P)^2 = e(P, P + P) = e(P, 2P)$$

or $\quad e(2P, P) = e(P, 2P)$

which implies that $\quad e(aP, bP) = e(P, P)^{ab} = e(abP, P) = e(P, abP)$

Another property of pairing is the *non-degeneracy* property, which states that the mapping of points does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity, or 1, in $\mathbb{G}_2$, i.e. $e(P, P) \neq 1$.

Fundamental Terminologies

Polynomials on Elliptic Curves: Let E be an elliptic curve over field K, then the set of defining polynomials on E is denoted by $K[X, Y]$. Any polynomial f on E can be written in the form $f(x, y) = v(x) + yw(x)$. These polynomials could be, for example, a line between two points or a vertical line through a point.

Rational Function: A rational function on E is an equivalence class of formal quotients of polynomials f/g (with g not identically zero).

Rational Function at P: If r is a rational function on E, and P is a finite point in E, then r is finite at P if there exists a representation of $r = f/g$ where f and g are polynomials on E, and f and g have some finite value at P but $g(P) \neq 0$. It is represented as $r(P) = f(P)/g(P)$.

Evaluating Rational Functions at $\mathcal{O}$: It is somewhat more complicated to define the value of a rational function at $\mathcal{O}$, even when it exists. The usual way to find the value of a rational function at infinity is to compare the degrees of the numerator and

the denominator. Let $r = f/g$ be a rational function with polynomials f and g. Since we can write any polynomial in the form $f(x, y) = v(x) + yw(x)$ then the degree of f is defined as:

$$deg(f) = max[2.deg_x(v), 3 + 2.deg_x(w)]$$

If $deg(f) < deg(g)$ then $r(\mathcal{O}) = 0$
If $deg(f) > deg(g)$ then $r(\mathcal{O}) = \infty$
If $deg(f) = deg(g)$ then $r(\mathcal{O}) = a/b$
where a and b are the coefficients of their leading terms ax^d and bx^d, respectively.

Poles and Zeros: Let r be a rational function on E. Then r is said to have a zero at a point $P \in E$ if $r(P) = 0$. If $r(P) = \infty$ then it is said to have a pole at P.

OR

If the order of a function f at point P i.e. $Ord_P(f) > 0$ then P is called the zero of the function f and if the $Ord_P(r) < 0$ then it is termed a pole of the function. The order of the point is described in the following definitions.

The Order of a Point and Torsion Points: The smallest scalar value for which a point is transformed to $\mathcal{O}$ (the point at infinity) is called the order of the point. All those points transformed to $\mathcal{O}$ by the order, say r, are called r-torsion points or killed-by-r-points.

Divisors: To keep track of the zeros and the poles of a rational function, one idea is to keep a list of them as:

$$[(P_1, m_1), (P_2, m_2), ..., (P_n, m_n)]$$

where m_i is the order. But another useful way to consider is their formal sum:

$$m_1 < P_1 > + m_2 < P_2 > + ... + m_n < P_n > = \sum_{i=1}^{n} m_i < P_i >$$

For example, $A = 3 < P_1 > -2 < P_2 > - < P_3 >$ is a divisor.

Degree of the Divisor: If D is a divisor,

$$D = \sum_{P \in E} m(P) < P >$$

$$\text{then} \qquad deg(D) = \sum_{P \in E} m(P)$$

If $deg(D) = 0$ then D is said to be a divisor of degree 0.

Divisors of Functions: Let r be a function on the curve E. Its divisor is denoted by $div(r)$, and is represented as:

$$div(r) = \sum_{P \in E} Ord_P(r) < P >$$

For example, let $ax + by + c = 0$ be the line passing through the points $P_1, P_2 \in E$, with $P_1 \neq \pm P_2$. This line intersects the curve at a third point $P_3 \in E$. Then the function $f(x, y) = ax + by + c$ has three zeroes P_1, P_2, P_3 and a pole of order 3 at infinity. The divisor of f is: $(f) = (P_1) + (P_2) + (P_3) - 3(\mathcal{O})$.

Principal Divisors: Let A be a divisor such that $A = \sum_P m(P) < P >$. If $\sum_P m(p) = 0$ and $\sum_P m(P) < P >= \mathcal{O}$ then A is a principal divisor and there exists a function f such that $div(f) = A$. We say that f is a function of the principal divisor A.

Evaluating a Function at a Divisor: If r is a rational function and A is a divisor then r is evaluated at A as:

$$r(A) = \prod_P r(P)^{m_P}$$

Computing the Weil Pairing:

Let m be a positive integer coprime to p, and $P, Q \in E[m]$. Let A_P and A_Q be divisors of degree 0 such that $A_P \sim < P > - < \mathcal{O} >$ and $A_Q \sim < Q > - < \mathcal{O} >$, and A_P and A_Q have disjoint support i.e their elements(points) are different (except for the point at infinity). Their corresponding functions f_P and f_Q exist such that

$$div(f_P) = mA_P$$

$$\text{and} \quad div(f_Q) = mA_Q$$

The Weil pairing of two points P and Q is computed as:

$$e_m(P,Q) = \frac{f_P(A_Q)}{f_Q(A_P)}$$

Computing of the Weil pairing is the computing of the corresponding rational function of the principal divisor with point P and evaluating it at the principal divisor with point Q and vice versa.

Computing the Function of a Principal Divisor: Any divisor of degree 0 can be written in its *canonical* form as:

$$D =< P > - < \mathcal{O} > + div(f)$$

where f is a rational function defined at points involved in the computation of the divisor D.
Let D_1 and D_2 be two divisors:

$$D_1 =< P_1 > - < \mathcal{O} > + div(f_1)$$

$$D_2 =< P_2 > - < \mathcal{O} > + div(f_2)$$

$$\text{then} \quad D_1 + D_2 =< P_3 > - < \mathcal{O} > + div(f_1 f_2 f_3)$$

where $P_3 = P_1 + P_2$, and $f_3 = l/v$ where l is the equation of the line through P_1 and P_2, and v is the equation of the vertical line through P_3.

Computing the rational function for a divisor is based on the canonical form above and the addition process. If m is the degree of the point P in divisor A_P then its function is computed by starting from 1 to m as the multiplicity of the divisor and for each subsequent stage a function like f_3 is computed and multiplied with the previous computed

ones. For further explanation, two examples on the computation of the Weil pairing are given in Appendix B.

The Weil pairing is computed by using Miller's algorithm [15] as described in the following.

Miller's Agorithm

Function: $D(f_b(A_Q), f_c(A_Q), bP, cP, (b+c)P) = f_{b+c}(A_Q)$.

(1) Let $a_1x + b_1y + c_1 = 0$ be the line passing through the points bP and cP (if b=c then let $a_1x + b_1y + c_1 = 0$ be the line tangent to E at bP). Define $g_1(x, y) = a_1x + b_1y + c_1$.
(2) Let $x + c_2 = 0$ be the vertical line passing through the point $(b + c)P$. Define $g_2(x, y) = x + c_2$

$$f_{b+c}(A_Q) = f_b(A_Q) - f_c(A_Q) - g_1(A_Q)/g_2(A_Q)$$

Main algorithm:
Let $n = b_m b_{m-1} ... b_1 b_0$ be the binary representation of n, i.e. $n = \sum_{i=0}^{m} b_i 2^i$.
Initialization: Set $Z = \mathcal{O}, V = f_0(A_Q) = 1$, and $k = 0$.

Iterate: For i = m, m-1 ..... 1, 0 do:
1: If $b_i = 1$ then do: Set $V = D(V, f_1(A_Q), Z, P, Z+P)$, set $Z = Z+P$, and set $k = k+1$.
2: If $i > 0$ set $V = D(V, V, Z, Z, 2Z)$, set $Z = 2Z$, and set $k = 2k$.
3: Observe that at the end of each iteration we have $Z = kP$ and $V = f_k(A_Q)$.

Output: After the last iteration we have $k = n$ and therefore $V = f_n(A_Q)$ as required.

Pre-computation:
For $f_1(A_Q)$ the function $f_1(x, y)$ (whose divisor is $(f_1) = (P + R_1) - (R_1) - (P) + (\mathcal{O})$) can be written explicitly as follows:

(1) Let $a_1x + b_1y + c_1 = 0$ be the line passing through the points P and R_1. Define the function: $g_1(x, y) = a_1x + b_1y + c_1$.

(2) Let $x + c_2 = 0$ be the vertical line passing through the point $P + R_1$. Define the function: $g_2(x, y) = x + c_2$.

(3) The function $f_1(x, y)$ is simply $f_1(x, y) = g_2(x, y)/g_1(x, y)$ which is easy to evaluate at A_Q in $\mathbb{F}_{p^2}$

3.2.2 The Tate Pairing

Another method for pairing of points with the bilinearity property is the Tate pairing. The Tate pairing is faster than the Weil pairing and is more often used in practice. The point P is chosen from $\mathbb{F}_p$, and Q from $\mathbb{F}_{p^2}$. Tate pairing is computed as:

$$e(P, Q) = f_p(A_Q)^{(p^2-1)/q}$$

Note that, as opposed to the Weil pairing, the evaluation of the divisor at the rational function is performed only once in the Tate pairing. Also, there are other techniques to make the computation faster in the Tate pairing, for example, by selecting a point Q of the form $Q[(a, 0), (0, d)]$ that stays in the same form under point multiplication. This results in a reduction of computational requirement. The pairing of points can be computed using Miller's algorithm described above. Further details of the selection of points can be learned from [34].

3.3 Conclusion

In this chapter, we described the identity-based cryptosystem proposed in [15]. We have also explained the computation methods for the pairing of points with examples, and gave basic terminologies to explain the technical details. The preferred method for computing the pairing of points is the Tate pairing due its efficiency. In our implementation of the proposed solutions of this thesis, we used the cryptographic library MIRACL that also uses the Tate pairing for faster implementation. However, further research is still underway to improve the computational methods for pairing operations to make pairing-based

cryptography more practical. For example, Barreto *et al.* in [39] have improved the performance of Tate pairings up to 10 times, and comment that their work can be extended to hyperelliptic curves, which can result in smaller key sizes and better performance.

Chapter 4

Authenticated Pairwise and Broadcast Protocols

In this chapter we explain the two security schemes we proposed for mobile ad hoc networks. One is identity-based with conventional key escrow that we call the basic scheme, and the other is key escrow free with the trade off of some identity-based features. Each scheme consists of authenticated pairwise and broadcast protocols.

These solutions use pairwise symmetric keys that are computed non-interactively by the nodes, which reduces communication overhead. We use identity-based keys that do not require certificates and simplify key management. Our key escrow free scheme also uses identity based keys but eliminates inherent key escrow problems. Our pairwise key protocol requires a minimum number of keys, $O(N)$, to be generated by the third party as compared to the conventional pairwise schemes with $O(N^2)$. We also propose an identity-based signature scheme for authenticated broadcast protocol. We allow nodes to generate their broadcast keys for different groups and propose a collision-free method for computing such keys.

For the authenticated broadcast protocol, we also propose a signcryption scheme as an alternative to our signature-encryption algorithm. It provides both signature and encryption operations in one algorithm, and an implicit control of the TA over the broadcast keys

generated by the users themselves. Finally, we give an overview of the research related more closely to the problems we consider in this chapter, than those papers discussed in chapter 1.

4.1 Introduction

As nodes in the ad hoc network are usually resource constrained, implementation of security solutions becomes more challenging. This implies that schemes which may provide strong security, may create severe computational and memory overhead, such as those using public key cryptosystems [4, 10]. On the other hand, some authentication schemes that require less computation [52, 51], either provide limited security or are difficult to implement in practice. Furthermore, most of the secure routing protocols focus on authentication. In order to ensure reliable network communication, authentication and non-repudiation are among the most essential requirements, not only to ensure reliable messages but also to identify malicious nodes. However, confidentiality and data integrity are also crucial requirements for a secure system. In most existing secure protocols for ad hoc networks, these mandatory requirements have either been ignored or only partially considered and require further setup. Our goal is to propose new solutions which provide all such mandatory features without any significant increase in computational, memory or bandwidth overhead.

The next sections are organized as follows: in section 4.2 we discuss an overview of our contribution and related security problems. In section 4.3, we describe our basic scheme which includes the description of the pairwise keys, broadcast keys, and the signature scheme. Section 4.4 describes our key escrow free scheme. We discuss performance and security analysis in section 4.5, and the signcryption scheme for the authenticated broadcast protocol is given in section 4.6. A brief survey of related work is discussed in section 4.7, followed by future work and conclusions in section 4.8.

4.2 Overview

In this section we give an overview of our proposed schemes and the related problems on which the security of our system is based.

4.2.1 Our contribution

We use the identity-based pairwise symmetric keys proposed in [40] for the authenticated pairwise communication in our basic scheme. Since the use of pairwise communication creates additional bandwidth overhead for broadcast messages, we propose an authenticated broadcast scheme that uses symmetric broadcast keys. As our broadcast keys are also symmetric, it lacks non-repudiation and will allow impersonation attacks. Thus, to ensure non-repudiation and to deal with impersonation attacks, we propose a signature scheme based on the broadcast secret. We allow nodes to generate their broadcast keys for different groups and propose a collision-free method for computing such keys. We also propose a key escrow free scheme that can be used with our proposed signature scheme for authenticated broadcast protocol. The pairwise key agreement in this scheme, like the basic scheme, is non-interactive, and requires less computation than the one in [46].

We also propose an identity-based signcryption scheme as an alternative to our signature-encryption algorithm for the broadcast protocol. It provides confidentiality and authentication by combining the two steps of signature and encryption into one singcryption algorithm, and also provides the TA with implicit control over broadcast keys despite their being created by the users themselves.

We assume that a semantically secure symmetric cryptosystem, such as AES, will be used for encryption and decryption once the secret keys are computed. The following are the related problems on which the security of our system is based.

4.2.2 A variant of Computational Diffie-Hellman Problem (CDHP)

The Computational Diffie-Hellman problem is defined as, given the points $< P, aP, bP >$, it is hard to compute abP. The security of our signature scheme is based on a variant of CDH problem also called the Inverse CDH problem (InvCDH), which can be defined as, given the points $< P, aP, bP >$, it is hard to compute $a^{-1}bP$. The proof of the security of InvCDH problem is given in [20].

4.2.3 Elliptic Curve Discrete Log Problem (ECDLP)

The Elliptic Curve Discrete Log Problem (ECDLP) is defined as, given a point P of order n on an elliptic curve over the field $\mathbb{F}_q$ and a point aP for some $a \in \mathbb{Z}_n$, it is hard to compute a.

4.2.4 Decisional Diffie-Hellman Problem (DDHP)

In our signature scheme, the Decisional Diffie-Hellman problem should be easy to solve in $\mathbb{G}_1$ in order to make the signature verification process possible. It is defined as, given $< P, aP, bP, cP >$, decide whether $c = ab \in \mathbb{Z}_q$.

Next, we define our basic scheme describing the computation of symmetric keys and the signature scheme.

4.3 Basic scheme

Let E be an elliptic curve $y^2 = x^3 + 1$ over $\mathbb{F}_p$, where $p = 2 \ mod \ 3$, and $p = lq - 1$ for some prime $q > 3$. Also q^2 should not divide $p + 1$. Let $\mathbb{G}_1$ be an additive subgroup of points on $E(\mathbb{F}_p)$ and $\mathbb{G}_2$ be the multiplicative subgroup on $E(\mathbb{F}_{p^2})$, both of order q. Also, let the point P be an arbitrary generator of $\mathbb{G}_1$.

Next, we describe the MapToPoint algorithm [15] that will be used to compute identity-based keys. This algorithm was also described in chapter 3, but, since we will be referring

it frequently in this chapter, we include it for the purpose of completeness.

The mapping function: MapToPoint

As described in [15], the MapToPoint function encodes the identity of the user to a point on an elliptic curve as follows.

We define the hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{F}_p$, which takes an input ID of any length and the resulting value is an element in the field $\mathbb{F}_p$. Let $y_0 = H_1(ID_X)$, where ID_X is the identity of any user X. The MapToPoint algorithm works as follows on the input $y_0 \in \mathbb{F}_p$:

(1) Compute $x_0 = (y_0^2 - 1)^{1/3} = (y_0^2 - 1)^{(2p-1)/3} \in \mathbb{F}_p$.

(2) Let $Q = (x_0,\ y_0) \in E(\mathbb{F}_p)$, and set $Q_{id\_X} = lQ \in \mathbb{G}_1$.

(3) Output $\mathbf{MapToPoint}(y_0) = Q_{id\_X}$.

The resulting point Q_{id_X} is the identity-based public key of the user X. We now describe the pairwise and broadcast communication protocols.

4.3.1 Authenticated Pairwise Communication Protocol

In this protocol, we compute pairwise shared keys through pairing of points, as described in [40]. After computing shared keys, any semantically secure symmetric cryptosystem, such as AES, can be used for the encryption and decryption processes. Authentication of such communication is ensured by the fact that the shared secret is only known to the two persons communicating (assuming neither is compromised).

Pairwise key computation

We assume that every node receives its private key based on its identity from the Trusted Authority (TA). In order to minimize key escrow (inherent in identity-based cryptosystems) and to achieve additional security for the master key s of the TA, we suggest the use of a threshold cryptography technique for s. Hence for any *node* A, its private key D_A will be computed as:

$$D_A = s_1 Q_{id\_A} + s_2 Q_{id\_A} + \cdots + s_n Q_{id\_A}$$

where $s_1, s_2, \cdots, s_n$ are the master keys belonging to n different TAs, and Q_{id_A} is computed as: $Q_{id\_A} = MapToPoint(H_1(ID_A))$ as described above. All the nodes should be issued private keys similarly using the same master keys. *Node A* can then compute its shared key with a *node B*, as proposed in [40], as:

$$D_{AB} = e\left(sQ_{id\_A},\, Q_{id\_B}\right)$$

where $sQ_{id\_A} = (s_1 + s_2 + \cdots + s_n)Q_{id\_A} = D_A$ is the private key of *node A* and Q_{id_B} is the public key corresponding to the identity of *node B*. Similarly, B can compute:

$$D_{AB} = e(Q_{id\_A}, sQ_{id\_B})$$

and based on the bilinearity property both results can be commonly represented as:

$$D_{AB} = e(Q_{id\_A}, Q_{id\_B})^s$$

For computing the symmetric key that will be used with the suggested symmetric cryptosystem, we define a hash function $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^m$ which gives a hash output of m bits for some element of group $\mathbb{G}_2$, where m is the key size as per requirements. The resulting key is:

$$K_{AB} = H_2(D_{AB})$$

An important observation is that this key agreement is non-interactive and does not require the involvement of the TA after the private keys have been issued. However, for such non-interactive key computation, the identities must be known to the users. When the network has online access, such information about user identities can be made available online. Whereas for the offline network, we describe the following pairwise-key management protocol that reduces the overhead of the TA to $O(N)$, as compared to $O(N^2)$ in conventional solutions.

Pairwise-key management protocol

In this protocol we require that users are issued identities by the TA, and such identities should be unique allowing clear recognition of users. For example, a user John Smith should be issued an identity like "john_smith@company.com" instead of "jsmith@company.com", which can cause confusion with other similar identities, for example, Judy Smith. Moreover, we require the TA to include further information with the identities if there are any conflicts. We consider such issues as policy issues that are outside the scope of this thesis.

For the above pairwise key computation by users, we claim that the overhead for the TA can be $O(N)$, instead of $O(N^2)$ as in conventional solutions such as in [51]. We achieve such reduction in overhead in two cases: first, when the network is small and users are familiar with each other; second, when network is large and users are not familiar with each other.

In the first case, where the network is small and users are familiar with each other, we require the TA to distribute only the private keys to the respective users. In a network of N users, this will require the TA to generate N private keys and distribute each key to its respective user. Recall that, for pairwise key computation, we require another parameter that is the identity of the other user. Since under this assumption users are familiar to each other, they can use the known identities, such as email addresses, for computing the pairwise shared secrets. Thus, the TA is only required to compute N private keys and users can compute pairwise shared keys non-interactively by using the known identities.

In the second case, where the network is large and users might not be familiar with each other, we maintain complexity $O(N)$ by a small modification in the source routing protocol for ad hoc networks, such as DSR. In the source routing protocol, if a node X wants to communicate with node Y and does not have the route for node Y, it will broadcast a route enquiry request to its neighboring nodes. The intermediate nodes will append their addresses to the response to that request and will re-broadcast the enquiry packet to other nodes. Once the packet reaches node Y, it will send back the routing

path to node X, and it will also keep the reverse of that path with itself. After deciding the appropriate path, nodes X and Y will start communicating with each other on that path. For pairwise key computation for nodes in such large networks, we require the TA to issue only respective private keys to the nodes. When node X broadcasts its route request, we require the intermediate nodes to include their identities in addition to the regular response to such requests. Once the complete route response reaches node X and Y, they have all the identities of the intermediate nodes which they did not know before. Node X is also required to put its own identity in the route request, and intermediate nodes can keep this identity for any future interaction with node X. This procedure is illustrated in the figure 4.1.

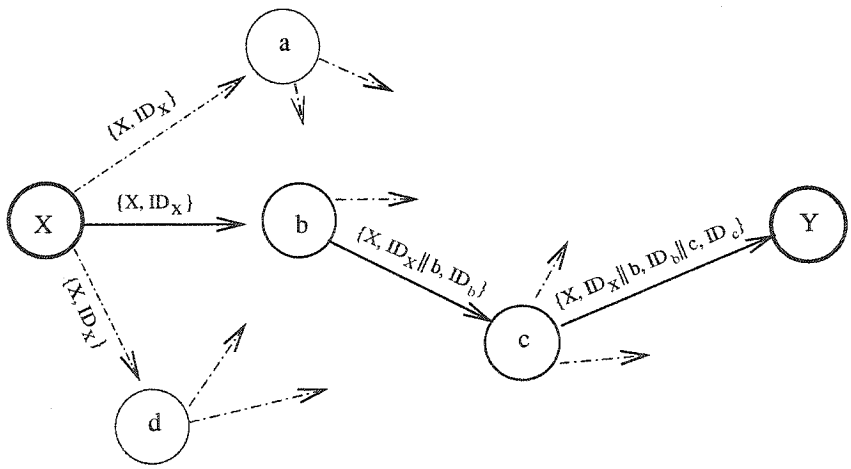


Figure 4.1: Route request broadcasted by node X

Since both nodes X and Y know the identities of the intermediate nodes, they can provide next-hop identities to the intermediate nodes in order to implement hop-by-hop authentication through pairwise encryption.

There are two advantages to this protocol: first, the work for the TA remains $O(N)$, and second, it helps to reduce the storage requirements for the nodes. It should be noted that, we do not require nodes to store pairwise keys for all the nodes in the network; instead nodes can generate and store only those keys that are in frequent use, and can delete the keys or identities that are not frequently used. Moreover, such discarded keys can always be recovered through the method described above. Thus, the storage requirement for the nodes is reduced. An important point also worth noting is that such key computation and deletion mechanisms are similar to the nature of source routing protocols in which nodes store only those routes that are in use or are frequently needed, and delete the unwanted routes. This makes source routing protocols more appropriate for ad hoc networks due to lower communication and storage requirements. In the same way, the above mentioned identity-based pairwise keys require less communication due to non-interactive computation, and reduce storage due to the protocol above.

It is important to mention here the major differences between identity-based pairwise keys and the conventional pairwise key computation through the Diffie-Hellman protocol.

- In the Diffie-Hellman key agreement protocol (described in Appendix A), two users pick their random secrets, and exchange public keys which are also based on those random secrets. Such random values can also be used by any intruder, which makes that protocol vulnerable to the *man-in-the-middle* attack. Whereas in ID-based pairwise keys, the public keys are based on identities that are issued by a third party and are not random. A person claiming an identity must also have the corresponding private key, which is associated with that identity and has been approved by the TA. Thus, an intruder claiming an identity cannot compute pairwise shared key without the corresponding private key.

- In the Diffie-Hellman protocol, even if a shared secret is computed, the authentication of the communicating parties is not confirmed. Public key certificates from a trusted authority are required to authenticate their identities. Such processes require additional communication between the nodes or the involvement of the Certification Authority (CA).

In case of ID-based pairwise keys, the computation of the shared keys implicitly authenticates the corresponding users and does not require the involvement of the TA after issuing the private keys. However, the conditions mentioned at the beginning of this protocol, regarding the identities, should be carefully implemented by the TA in order to avoid any confusion or impersonation attempts.

- In the Diffie-Hellman protocol, users can generate as many pairwise keys as they need, whereas in the ID-based system they need to get their identities changed or modified by the TA to generate different shared keys. However, the Diffie-Hellman protocol can be incorporated in the ID-based system to generate new shared keys after verifying the authentication of users through ID-based shared keys, which requires neither certificates nor the involvement of the TA after issuing the private keys.

Next, we describe the authenticated broadcast protocol.

4.3.2 Authenticated Broadcast Communication Protocol

Since the use of pairwise communication in broadcast messages consumes additional bandwidth, we use symmetric broadcast keys to minimize such overhead, and a signature scheme for authentication. Next, we describe the computation of broadcast keys.

Broadcast key computation

Let P be a point in the group $\mathbb{G}_1$, and K_{1N} be the broadcast secret of a *node 1* for any group of N nodes. *Node 1* will compute its broadcast parameter P_{1_brdcst} as:

$$P_{1\_brdcst} = K_{1N} \cdot P$$

and distribute it to all candidate nodes using respective pairwise encryption. Every node will then compute the broadcast key of *node 1* as K_{1_brdcst} using the hash function $H_3 : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^m$ which outputs a key of size m from the input of two elements of group $\mathbb{G}_1$. The key K_{1_brdcst} is:

$$K_{1\_brdcst} = H_3(P_{1\_brdcst})$$

NOTE: The broadcast parameter P_{1_brdcst} will be used to verify signatures as shown in the next section, and the key K_{1_brdcst} will be used for encryption by *node 1* and for decryption by other nodes. The key K_{1_brdcst} can also be computed by *node 1* and delivered with P_{1_brdcst}. We leave the choice of approach as part of network policy.

For computing the broadcast secret K_{1N}, we describe here two different methods:

(i) K_{1N} can be any random value generated by *node 1* and used as its broadcast secret. This approach of using random secrets is commonly used in practice, since if a node leaves the group, the broadcast key can be changed dynamically and delivered to the desired group of nodes. The node which has quit the group will no longer be able to access the broadcast messages of *node 1*.

(ii) When large networks and smaller fields are used, the probability of having any two nodes at any time getting the same random value is negligible but not precisely zero. Hence, if it is required that every node must have a unique key, the following method ensures that K_{1N} is the unique key that only *node 1* can compute, unless all the other $n-1$ nodes collude. Such a broadcast secret can be computed for different groups and can be changed accordingly. This method will ensure collision-free keys provided each member has a unique identity. We compute K_{1N} as follows:

$$\begin{aligned} D_{1N} &= e\,(sQ_{id\_1}\,,\; Q_{id\_2} + Q_{id\_3} + \cdots + Q_{id\_n}) \\ &= e\,(sQ_{id\_1}\,,\; Q_{id\_2}) \cdot e\,(sQ_{id\_1}\,,\; Q_{id\_3}) \cdots e\,(sQ_{id\_1}\,,\; Q_{id\_n}) \\ &= D_{12} \cdot D_{13} \cdots D_{1n} \end{aligned}$$

and $\quad K_{1N} = H_2(D_{1N})$.

It is recommended that the secret K_{1N} be at least 160 bits long for the discrete log problem to be hard enough. The above broadcast key is safe against the collusion attack within the group, but if there is any procedure of accountability to a higher level authority then such collusion can be effective against a user as the collusion of all group nodes can compute the broadcast secret of *node 1* and can generate false messages with the signature of *node 1*. In order to avoid such problem, we require the TA to distribute a point P' to all the nodes in the network and using that point, nodes will compute their broadcast keys as:

$$D_{1N} = e\left(sQ_{id\_1}\,,\; Q_{id\_2} + Q_{id\_3} + \cdots + Q_{id\_n} + P'\right)$$

Since, even the collusion of all other nodes in the group cannot compute the pairing $e\left(sQ_{id\_1}\,,\; P'\right)$, we can easily avoid such attacks as well.

Application of collision-free broadcast key computation: The above mentioned collision-free broadcast key computation can be useful in environments like military deployment, where key conflicts due to the probabilistic approach cannot be afforded due to the limited time of operation and the critical nature of the network. Moreover, unique key computations should not cause any additional information exchange among the nodes. Our solution satisfies both criterion, provided the identities of the group members are known and each member is assigned a unique identity by the TA. Under that assumption, such key computation is non-interactive and will not cause additional communication.

Next, we describe the signature scheme based on the broadcast secret, in order to ensure authentication and non-repudiation in the broadcast protocol.

Signature scheme

Since we are using symmetric broadcast keys which do not ensure non-repudiation and are vulnerable to impersonation attacks, we propose a signature scheme based on the broadcast secret described above. According to this scheme, *node 1* will compute the inverse

of its broadcast secret as K_{1N}^{-1} which will be used in signature generation as follows.

Signature generation

For generating a signature, *node 1* will compute $h = H_4(M)$, where H_4 is defined as $H_4 : \{0,1\}^* \rightarrow \{0,1\}^m$ for a message M of any length and results in a hash of m bits. A random value r is generated where $r \in \mathbb{Z}_q^*$. A parameter U is computed as: $U = rQ_{id\_1}$, and $V = K_{1N}^{-1}(r+h)Q_{id\_1}$. The signature σ consists of two elements:

$$\sigma = \{U\ ,\ V\}$$

$$= \{rQ_{id\_1}\ ,\ K_{1N}^{-1}(r+h)Q_{id\_1}\}$$

Signature verification

For verification, any node can compute $h = H_4(M)$ and $Q_{id\_1} = MapToPoint(H_1(ID_1))$, and check if:

$$e(P_{1\_brdcst}\ ,\ V) = e(P\ ,\ U + hQ_{id\_1})$$

This follows from the fact that:

$$e(P_{1\_brdcst}\ ,\ V) = e(K_{1N}P\ ,\ K_{1N}^{-1}(r+h)Q_{id\_1})$$

$$= e(P\ ,\ (r+h)Q_{id\_1})^{K_{1N}\cdot K_{1N}^{-1}}$$

$$= e(P\ ,\ (r+h)Q_{id\_1})$$

and $\quad e(P\ ,\ U + hQ_{id\_1}) = e(P\ ,\ rQ_{id\_1} + hQ_{id\_1})$

$$= e(P\ ,\ (r+h)Q_{id\_1})$$

This signature scheme can also be used for our key escrow free scheme that we define in the next section.

4.4 Key escrow free scheme

In order to work around key escrow inherent in identity-based keys, we propose a modification to the scheme proposed in [46]. We compute the keys for the symmetric cryptosystem, whereas the pairwise keys in [46] are computed from their proposed public key cryptosystem and thus require more computation. We present two variations of our scheme based on group identity and individual identities.

4.4.1 Group identity-based

A group public key Q_{ID} is to be generated by the TA based on any group identity or arbitrary string. The TA, using its master key s, then computes the initial group key:

$$D = s \cdot Q_{ID}$$

Every *node i* will then receive the point D from the TA and will generate its private key k_i, a random secret, and compute the corresponding public key as:

$$D_{i\_pub} = k_i \cdot D \qquad \text{for} \quad 1 \leq i \leq n$$

All such individual public keys should be provided to the TA. The participating nodes will then get the public key of every node from the TA.

We then compute the pairwise shared secret between any two nodes without transferring any information between the two nodes as follows.

A *node 1* using its private key k_1 and the public key of a *node 2*, i.e. $k_2 D$, can compute $k_1 k_2 D$, and similarly *node 2* will use its private key k_2 and the public key of *node 1* and

will compute $k_2 k_1 D$. The *nodes 1* and *2* will then compute:

$$K_{12} = H_3(k_1 k_2 D)$$

Each pair of nodes can compute their pairwise secret key as described for *nodes 1* and *2*. This key agreement is also non-interactive, like our basic scheme.

For the broadcast key, the parameter $P_{1\_brdcst} = K_{1N} \cdot P$ is computed as in the basic scheme, with K_{1N} being any random secret. The signature scheme would be used as described in the basic model.

This scheme not only does not use certificates, but also removes the key escrow inherent in the conventional identity-based cryptosystem. But the restriction in this group identity-based scheme is that all the nodes will have their pairwise keys expired at the same time since they are using one common Q_{ID} from the TA. However, this could be a useful restriction in certain environments and applications; for example, users gathered at some place for conference can be issued such keys only for the duration of the conference.

Next, we describe the key escrow free scheme based on individual identities.

4.4.2 Individual identity-based

Based on the individual identity, the TA will compute the partial private key of a *node 1* as:

$$D_1 = s \cdot Q_{id\_1}$$

The *node 1* will then compute its private key as:

$$k_1 = H_3(D_{1x})$$
$$\text{where} \quad D_{1x} = x_1 \cdot D_1$$

and x_1 is a random secret chosen by *node 1*. *Node 1* will then compute its public key as:

$$D_{1\_pub} = k_1 \cdot P$$

and will submit it to the TA. The pairwise and broadcast keys will be computed similarly, as discussed in section 5.1.

4.5 Analysis

In this section we discuss the performance and security analysis of the above schemes.

4.5.1 Performance analysis

In our schemes, we are using pairwise symmetric encryption for authentication, which is faster and requires less computation than public key cryptosystems. Our basic scheme starts with only private keys to be issued to the nodes and does not require a large number of pairwise keys to be generated by a third party. In the conventional pairwise key scheme proposed in [51], a third party is required to generate $n(n-1)/2$ keys for n nodes, and provide every node with $(n-1)$ keys, necessitating $n(n-1)$ keys to be distributed through some secure channel. However, our scheme needs only n private keys to be generated by the TA who then provides the respective private key to each of the n nodes. Every node can then compute a pairwise shared secret for any other node at any time without further exchange of information, provided the identities (such as email addresses, student or employee IDs etc.) are publicly available or collected through the pairwise-key management protocol described above. Our basic scheme requires less storage as it does not require a node to store keys for all the other nodes. In fact, a node can delete the keys that are not in frequent use and can generate such keys any time later without contacting the TA. This approach can save memory and provide dynamic security as required.

Our signature scheme is a modified form of the one in [26] with different parameters and operations being used. We have one more point multiplication in the signature generation in parameter $V = K_{1N}^{-1}(r+h)Q_{id\_1}$. However, this additional computation can be alleviated by taking the multiplication of $K_{1N}^{-1}.Q_{id\_1}$ only once and using it for all other broadcast messages since it is independent of the message being signed. Hence, we claim that our scheme is as efficient as [26], which is considered to have the least computational

overhead of all the currently proposed identity-based signature schemes.

In the following table, we give the approximate computation time required in the signature generation and verification algorithms of our signature scheme and the one in [26]. We implemented these algorithms on a Pentium II machine with the CPU speed of 333 MHz, 192 MB RAM, and the Redhat 2.4.18 Linux operating system.

	Proposed signature		Signature in [26]	
	Gen. alg. (in secs.)	Verif. alg. (in secs.)	Gen. alg. (in secs.)	Verif. alg. (in secs.)
With pre-computation	0.094140	0.605951	0.094140	0.605951
Without pre-computation	0.136584	0.594815	0.116263	0.594815

The use of signatures in broadcast messages increases computational overhead. However, it requires less bandwidth than when broadcast messages are sent separately for each node using pairwise encryption. Our key escrow free pairwise keys need less computation than those in [46].

4.5.2 Security analysis

The security of our system is based on the discrete log problem and a variant of Computational Diffie-Hellman problem. In order for the discrete log problem to be hard enough, it is recommended that the broadcast secret, K_{1N} be at least 160 bits long, and the elements of $\mathbb{G}_1$ be at least 512 bits long. In our scheme, if a node is compromised or an authorized node is malicious, all the pairwise keys associated with that node and its broadcast key could be misused. However, in such cases the broadcast keys of other nodes cannot be misused due to the signature scheme. Methods of detecting the malicious behavior of such nodes are outside the scope of this paper. However, due to our authentication and non-repudiation features, the source can always be identified. On the other hand, the passive attack of eavesdropping by such nodes is difficult to detect but can be limited through hop-by-hop authentication and encryption using pairwise shared keys. Our scheme can

also implement any efficient Message Authentication Code (MAC) schemes to ensure data integrity, since we are using symmetric keys.

The security of our signature scheme is based on a variant of the Computational Diffie-Hellman (CDH) problem also called the Inverse CDH problem (InvCDH), which can be defined as, given the points $< P, aP, bP >$, it is hard to compute $a^{-1}bP$. This is evident from our signature generation algorithm i.e. if an attacker can compute the parameter $K_{1N}^{-1}(r+h)Q_{id\_1}$, given $(r+h)Q_{id\_1}$ and $K_{1N}P$, then the attacker can solve the variant of the CDH problem and can sign any messages on behalf of any user. The above mentioned parameters of the signature scheme comply with the definition of the variant of the CDH problem due to the fact that, since P is the generating point of the curve, Q_{id_1} will be some multiple of P, say $Q_{id\_1} = aP$. Then the above statement can be re-written as: if an attacker can compute the parameter $K_{1N}^{-1}(r+h)aP$, given $(r+h)aP$ and $K_{1N}P$, then the attacker can solve the variant of the CDH problem and can sign any messages on behalf of any user. Thus, the security of our signature scheme is based on the variant of the CDH problem, the InvCDH problem. But the InvCDH problem is considered hard, and a proof of it is given in [20]. Hence, we claim that our signature scheme is secure, since the InvCDH problem is proven to be difficult to solve in the group $\mathbb{G}_1$.

As an alternative to our signature-encryption algorithm for authenticated broadcasting, we propose an identity-based signcryption algorithm that is discussed in the next section.

4.6 Signcryption scheme

We propose a pairing-based signcryption scheme for authenticated broadcasting, which requires less computation than other proposals with similar objectives, such as [4]. Due to the dynamic nature of ad hoc networks, we allow nodes to generate their own broadcast keys for different groups in the network and to change them when the associated groups are changed. In this protocol, we ensure, through our signcryption scheme, that such broadcast keys would be implicitly controlled by the Trusted Authority (TA), and can

be used for as long the private keys are valid. This control in our signcryption scheme is an additional feature to our previously proposed authenticated broadcast protocol with signature-encryption scheme.

Next, we describe the computation and distribution of broadcast keys in the initialization step.

4.6.1 Initialization

Let Q_{id_1} be the public key assigned to *node 1* by the TA, and K_{1N} be the broadcast secret of *node 1* for a group of N nodes. *Node 1* will compute the broadcast parameter P_{1_brdcst} as:

$$P_{1\_brdcst} = K_{1N} \cdot Q_{id\_1}$$

The broadcast secret K_{1N} can be computed through two methods that we have described above.

The secret K_{1N} should be long enough for the discrete log problem to be hard. *Node 1* will deliver the broadcast parameter P_{1_brdcst} to other users in group by encrypting in each member's pairwise shared key with *node 1*.

Next we describe our proposed signcryption scheme for the authenticated broadcast protocol.

4.6.2 Signcryption scheme

The signcryption scheme consists of two algorithms: Signcrypt and Unsigncrypt, described as below.

Signcrypt:

In order to signcrypt the message M, *node 1* will compute $h = H_3(M)$, where $H_3 : \{0,1\}^* \rightarrow \{0,1\}^*$. A random value $r \in \mathbb{Z}_q^*$ is generated, and a parameter d_1 is precomputed as follows and stored for the encryption of broadcast messages:

$$d_1 = e\,(Q_{id\_1}\,,\ P)$$

Message M is encrypted with key $K_{1\_brdcst} = H_2(\ d_1^{(r+h)}\ )$ as:

$$C = E_{K_{1\_brdcst}}(M) = M \oplus K_{1\_brdcst}$$

Two parameters U and V are computed as:

$$U = rP\ , \quad \text{and} \quad V = K_{1N}^{-1}\,(r+h)P$$

The broadcast message β consists of three elements:

$$\begin{aligned}\beta &= \{C\ ,\ U\ ,\ V\} \\ &= \{E_{K_{1\_brdcst}}(M)\ ,\ rP\ ,\ K_{1N}^{-1}\,(r+h)P\}\end{aligned}$$

Unsigncrypt:

For decryption of the message, the authorized receivers (members of the group provided with broadcast parameter $K_{1N}Q_{id\_1}$) will compute the key K_{1_brdcst} as a hash of d_2, where d_2 is computed as:

$$\begin{aligned}d_2 &= e\,(K_{1N}Q_{id\_1}\,,\ V) \\ &= e\,(K_{1N}Q_{id\_1}\,,\ K_{1N}^{-1}(r+h)P) \\ &= e\,(Q_{id\_1}\,,\ (r+h)P)\end{aligned}$$

and thus, $\quad K_{1\_brdcst} = H_2(d_2)$

Message M is decrypted as:

$$M = D_{K_{1\_brdcst}}(C) = C \oplus K_{1\_brdcst}$$

After decrypting message M, its hash can be computed as: $h = H_3(M)$, and authentication is verified by computing $Q_{id\_1} = MapToPoint(H_1(ID_1))$, and d_3 as:

$$d_3 = e\,(Q_{id\_1}\,,\; U + hP)$$

$$= e\,(Q_{id\_1}\,,\; rP + hP)$$

$$= e\,(Q_{id\_1}\,,\; (r+h)P)$$

The message is verified to be from *node 1* if d_3 is equal to d_2.

We claim that the broadcast keys are implicitly controlled by the TA; this is ensured through the signcryption scheme. It is to be noted that for computing parameter d_3 using unsigncrypt algorithm, receivers of the broadcast message will compute the public key Q_{id_1} for *node 1* using the identity for which the sender was issued a private key by the TA. If the broadcast key of *node 1* is computed as $K_{1N}Q'_{id\_1}$, e.g. by changing the validation date in ID, the verification by other nodes would fail and such broadcast messages would not be authenticated.

4.7 Comparison to previous work

Different security mechanisms have been adapted in different proposed solutions. In SEAD[52], the security scheme suggested is based on hash chains. A node will generate a hash chain as: $h_0, h_1, h_2, \cdots, h_n$, where $h_i = H(h_{i-1})$ for $0 < i \leq n$, for some n. In order to achieve authentication, the distribution of the hash element h_n is assumed to be through some trusted mechanism like a public key certificate signed by a trusted entity. Authentication is achieved as: given h_n, h_i can be verified by computing $h_n = H^{n-i}[h_i]$ for $i < n$. Use of hash chains makes this protocol much faster than public key cryptosystem based protocols. Since each hash element would only be used once, in case of increased network traffic, a significantly large number of hash elements is required. In Ariadne[51], h_n is distributed similarly as an authenticated key. There is no message integrity in SEAD,

whereas Ariadne uses a Message Authentication Code (MAC) for integrity. Ariadne uses clock synchronization and delayed key disclosure, and relies on the ability of users to determine which keys a sender may have already published. However, some features, like clock synchronization and the knowledge of end-to-end delay, are difficult to implement in mobile environments and there is additional communication overhead in implementing such a protocol.

In ARAN[10], security is based on public key certificates and it is assumed that keys are generated *a priori* and exchanged between a certificate server and nodes. Any entering node requests a certificate from the server and is then allowed to participate in the network. The suggested use of certificate based public key cryptography causes additional overhead for resource constrained nodes due to its computational and memory costs.

In [43], an accelerated scheme for the key establishment protocol is proposed which uses the Server-Aided Secret Computation (SASC) technique. The SASC technique (also used in [48, 12, 2]) exchanges information with the base station to get expensive computation done by the server. In such protocols, a prior arrangement with the base station is required which restricts the independence of nodes in return for assistance by the base station.

In [4], the distributed Certification Authority (CA) scheme proposed by Zhou and Haas [30] is improved by suggesting the use of Identity-Based Encryption (IBE) and an applicable signature scheme. The use of IBE removes the overhead of certificates. The authors propose the establishment of a distributed Private Key Generation (PKG) service within the network instead of a CA. The entering nodes get their private keys from the distributed PKG based on a t-out-of-n scheme. However, technical details for computing private keys (generated by t-out-of-n authorities) for the corresponding master public key are not given. It is also mentioned that the network initialization stage is vulnerable to Byzantine failures. The use of the suggested IBE and signature scheme is computationally more expensive than symmetric cryptosystems.

In [26], Cha and Cheon present an identity-based signature scheme. In this scheme the master public key $P_{pub} = sP$ is computed by the TA and distributed to all the users. A user using his private key $D_{ID} = sQ_{ID}$ will compute his signature σ for some message m, as: $\sigma = \{U, V\}$, where $U = rQ_{ID}$, $V = (r + h)D_{ID}$, $h = H(m, U)$, and r is a random value. The signature is verified if:

$$e(P, V) = e(P_{pub}, (r + h)Q_{ID})$$

As after receiving the message, other users can compute $h = H(m, U)$ and also hQ_{ID}, which can be used to compute $(r + h)Q_{ID}$ through $U + hQ_{ID}$. Our signature scheme has a similar structure, but different parameters and operations. We have one more point multiplication in the signature generation process, which can be alleviated as described above. Also, a difference can be noted at the cancellation of $K_{1N}K_{1N}^{-1}$ in our verification process. The signature scheme in [26] is based on the Computational Diffie-Hellman (CDH) problem, whereas ours is based on a variant of CDH problem called InvCDH problem as described above.

4.8 Conclusions and future work

We have presented two efficient identity-based security schemes that are based on pairwise symmetric keys. Since conventional identity-based keys suffer from the key escrow problem, we propose a key escrow free model as an alternative to our basic scheme if escrow free security is required. However, in our basic scheme we tend to minimize this problem using the threshold cryptography technique. The use of threshold techniques is natural in many environments. For example, an organization or institution can act as the Trusted Authority (TA) for its employees or students, and the master keys can be generated and stored distributively at different administrative levels. In a large group of nodes, it is natural that a certain subgroup of nodes will have more communication with each other and less or no communication with the nodes in other subgroups. Thus, they should not be issued unwanted keys for all the nodes. Our basic scheme does not require a large number of pairwise keys to be generated by a third party, unlike conventional

schemes. We need only the private keys to be issued by the TA, and nodes are free to generate shared secrets as needed, without contacting the TA. In order to provide such autonomous mechanisms, the required identities should be publicly available.

Since the use of pairwise communication creates additional bandwidth overhead for broadcast messages, we have proposed an authenticated broadcast scheme based on symmetric keys and a corresponding signature scheme. The use of signatures in broadcast messages increases computational overhead, but it ensures non-repudiation and minimizes bandwidth overhead. Since our signature verification process is computationally more expensive than signature generation, a possible attack by malicious nodes is to generate computational overhead for other nodes by sending unnecessary broadcast messages. Methods for analyzing such behavior are required to defend against such attacks. However, the non-repudiation and authentication in our scheme always identify the source of such attempts and help to deal with such attacks. We also propose an identity-based signcryption scheme as an alternative to our signature-encryption scheme. It has comparatively balanced operations in signcrypt and unsigncrypt algorithms, and provides implicit control to the TA over the broadcast keys generated by the users.

We have tested our proposed scheme using the MIRACL library with required modifications. Further, we will analyze the simulation of our scheme, implemented with DSR and/or DSDV protocols. In [39], P. Barreto *et. al.* have proposed a new pairing technique using non-supersingular curves and have improved the performance of the Tate pairing up to 10 times. It is suggested that their work could be extended to hyper-elliptic curves, which can result in smaller key sizes and improved performance. Such implementations would reduce the computational overhead of pairing, and would make pairing-based schemes, like ours, more practical.

Chapter 5

Self-healing in the Group Key Distribution

In network communication, group communication is often carried out using a group key. Such environments assume that users are trusted and will not impersonate others. Although, in practice, such assumptions are not realistic, but group keys can still be useful to secure group information for which source of origin is not important. For group key distribution and revocation in mobile ad hoc networks, we surveyed several proposals and found that the Subset Difference (SD) method proposed by D. Naor *et al.* is the most efficient group key distribution technique, even for large networks. Recently, a polynomial based solution for key distribution was proposed by D. Liu *et al.*, which requires a similar message size as the SD method, but also provides a self-healing feature. Since in mobile ad hoc networks, due to their wireless nature and mobility, there is more possibility of packet loss; we thus propose a self-healing feature (that allows nodes to recover lost keys) for the SD method to make it a more practical solution for ad hoc networks, and will show that the performance of the SD method is superior to the polynomial based solution. We also present some optimization techniques to reduce the overhead caused by the self-healing capability. Finally, we also propose the idea of mutual healing and mention certain requirements for this method of key recovery.

5.1 Introduction

In network communication, messages that are intended for more than one user should be delivered through multicasting to save network resources. There is a need to control access to the content of such messages, and to restrict them to authorized users only. However, in practice, the group of authorized users can vary periodically. Conventional solutions use group keys to implement such control, but face problems in dealing with changes in the group of authorized users. As the group size grows larger, scalability becomes an issue and more efficient protocols are required to provide a desired level of security without trading away performance.

In wireless mobile networks and networks with heavy traffic such as the Internet, packet loss, which can result in loss of crucial information by users, is common. For example, in a key revocation process in such networks, it is very likely that users might not receive their updated session keys. Thus, it is very important to ensure reliable transmission of updated session keys to the authorized users. One naive approach to solve this problem is to expect, those users who did not receive the message to contact the server to get the required key. However, in the case of very large networks, if a significant number of users approaches the server to get the lost session keys, the bandwidth overhead will increase and it will be difficult for the server to manage such requests. A more practical approach is to ensure self-healing in the broadcast messages of the updated session keys. This can reduce the communication overhead and can simplify key management for the server. One major problem in using a self-healing technique is that the broadcast message size will increase in proportion to the number of session keys which are made recoverable through self-healing. Also, the addition of new members or rejoining of old members requires the protection of previous session keys from unauthorized users. This increases computational requirements and the message size. Thus, optimization techniques are required to control the message size and computational requirements of any self-healing scheme.

In this chapter, we propose a self-healing technique for the SD key distribution scheme and will also give optimization methods that can be used to minimize message size and

computational requirements.

5.1.1 Related work

The first significant solution for broadcast encryption and revocation can be traced back to the broadcast encryption scheme by Fiat and Naor [3]. Their solution allows the key distribution center to broadcast messages to authorized users while removing a subgroup of users, and such revocation is secure against the collusion of a group of, say k, users. Another key distribution and revocation scheme proposed in [25], is based on the polynomial interpolation technique, which was also used in [33] and [23] for the revocation of users. The scheme in [25] has self-healing capability and keys are distributed to users in the group using a polynomial covered with a masking polynomial. In order to ensure self-healing, partial shares of other session keys are included in the message of each session key. It is sufficient for a user who has missed the key for some session j to have at least one message from the previous update messages before session j, and any one message after session j. Information from the two messages will be used to compute the key for session j. For this recovery, a user does not have to contact the group manager, rather he only has to listen to the traffic which should enable him to get such a key. However, the size of the broadcast message in this scheme increases exponentially as the number of revoked users increases, and its security against the collusion of revoked users is associated to the degree of polynomial.

In [18], a self-healing technique similar to that in [25] is used; however, the key distribution technique is different and more efficient. It can be shown that, by excluding the self-healing feature, its broadcast message size is a linear function of the number of revoked users and is very close to the message size of the SD scheme proposed in [1], which will be discussed in detail later in this section. The key recovery operation in [18] is more expensive than in [1]. The SD scheme is secure against the collusion of any number of revoked users, whereas [18] is limited to the degree of the polynomial being used. This will imply that as higher degree polynomials are used, the computation becomes more expensive. However, the scheme in [18] has the self-healing capability in addition to the

revocation process.

In [17], a Layered Subset Difference (LSD) scheme is proposed that reduces the initial key distribution requirement by almost a square root factor over the basic SD scheme. It uses unions of small subsets and reduces such collections of subsets, and thereby reduces the number of initial keys given to users. The authors in [17] also discuss more complicated types of privileged sets defined by nested inclusion and exclusion conditions. The storage requirement in both SD and LSD is further reduced in [47] with an increase in computational overhead.

The SD protocol is based on the idea of a logical key hierarchy (LKH) proposed in [19] and [14]. The LKH protocol uses key hierarchy instead of the hierarchy of group security agents. It is also a re-keying protocol like SD, but revokes a single user at a time and updates the keys of all remaining users. In [54], the authors use periodic batch rekeying (instead of rekeying after each join or leave as in LKH) to improve scalability. For reliable key transport, forward error correction (FEC) coding is used in [54]. C. Wong *et al.* in [13], use UDP over IP multicast for efficient rekey message delivery for large groups. In order to avoid the server being a bottleneck in the registration process, multiple registrars are used to offload client registration from the server. As the UDP protocol is unreliable, packet loss is possible. If the message is larger than the message transmission unit (MTU), IP fragmentation occurs and the message is broken into several IP packets; this increases the message loss probability. To reduce the message loss probability, the authors use the FEC technique. In the case of the message loss, re-synchronization of the lost keys to the requester is done by using pairwise encryption. In [5], the re-synchronization process is non-interactive due to the use of hints for updated keys attached to subsequent data packets. Such hints can be as small as half of the actual key size and users are required to use it to compute the key. The solutions in [19, 14, 13, 5, 54] are stateful i.e. if the keys in the tree are updated, an offline member will remain separated from the group and cannot rejoin without assistance from the group manager.

In [11], Pinkas proposes two solutions to reduce broadcast message size. In his first

scheme, a pseudorandom function is used and keys are generated using two seeds. A user remaining offline for a certain period will be given appropriate seeds to derive keys for that period. Since keys are not independent of each other, this scheme is vulnerable to the collusion attack by two users who can generate keys for the unauthorized sessions that are in between their authorized sessions. In the second solution, the session keys are derived from a pseudorandom function on the pattern of a binary tree. The leaves of the tree are considered to be the session keys. Any user remaining offline for a period of t consecutive sessions will be given the key of the root of the subtree containing all such sessions. The collusion of any number of users will not allow them to compute keys for the sessions from which they were excluded. This solution requires requests from the users (who have lost their updates) to the server to re-send the lost keys. Our focus is to avoid such requests and thereby reduce the possibility of the server being a bottleneck. We are also presenting a self-healing solution, which ensures that no user can use this capability to compute keys for the sessions for which the user was not authorized.

Next, we will discuss some of the details of the Subset Difference method proposed in [1].

5.2 Preliminary information

In conventional tree based schemes, a user is only considered to be the member of subtrees rooted at its ancestors, whereas in the SD scheme it can also be covered in subtrees not rooted at its ancestors. It was shown that this approach can reduce the overall number of constructed subsets needed to deliver the new group key while revoking certain users. The reduced number of such subsets eventually reduces the message size. A subset in SD is described below.

5.2.1 The subset description

A subset $S_{i,j}$ is represented by two nodes $(v_i,\ v_j)$, such that v_i is the ancestor of v_j and the subset $S_{i,j}$ consists of the node v_i and its descendants excluding the node v_j and its

descendants.

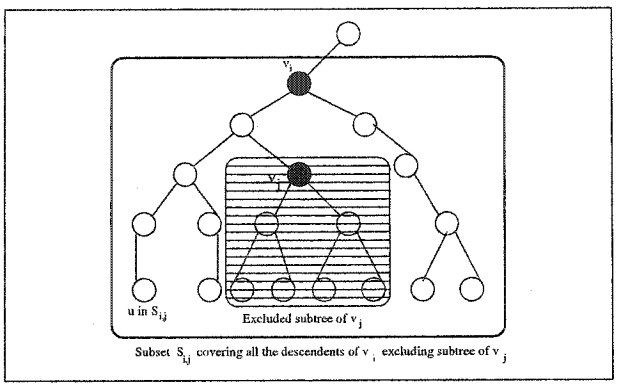


Figure 5.1: The Subset Difference Method: subset $S_{i,j}$

In this scheme, at an initialization step, a user is given the labels of those nodes that are adjacent to its ancestors but not the labels of its ancestors. A user can derive keys for other descendants from such labels. When a new group key is to be delivered, it is encrypted in the subset key in order to protect it from revoked users. A subset key is the key of the first common ancestor of revoked users that is derived from the label of the root of that subset. Thus, when some users are to be revoked from the group, subsets are constructed through the SD algorithm and a new group key is delivered by encrypting with each subset key for non-revoked users.

5.2.2 The Subset Difference (SD) method

For a given set $\mathcal{R}$ of revoked users, the non-revoked users of $\mathcal{N}\backslash\mathcal{R}$ are partitioned into subsets $S_{i_1,j_1}, S_{i_2,j_2}, \cdots, S_{i_m,j_m}$ as follows:

- A Steiner tree $ST(\mathcal{R})$ is constructed of revoked users and the root.
- The subset collection is obtained iteratively, maintaining a tree T which is a subtree of $ST(\mathcal{R})$. T is initially equal to $ST(\mathcal{R})$ and then nodes are removed iteratively from T through the following steps until it is reduced to a single node.

Step 1 : This step will find nodes with the following properties:

(a) IF: there is more than one leaf in the tree T

then: find two leaves v_i and v_j (that are to be revoked) in the tree T, such that the least-common-ancestor v of v_i and v_j does not contain any other leaf of T (i.e. no more than two members of $\mathcal{R}$). The node v would be of outdegree 2, i.e. it has two members of $\mathcal{R}$ on both its branches. Let v_l and v_k be the two children of v such that v_i is the descendant of v_l, and v_j is a descendant of v_k.

(b) else IF: there is only one leaf to revoke or after all iterations only one leaf is left in the tree T

then: the main root will be considered as $v = v_l = v_k$ and the node to be revoked can be considered either v_i or v_j.

Step 2 :

IF: $v_l \not\equiv v_i$ (i.e. the two nodes are different)

then: add subset $S_{l,i}$ to the collection of subsets,

similarly, IF: $v_k \not\equiv v_j$

then: add $S_{k,j}$ to the collection of subsets.

Step 3 : Remove from T all the descendants of v and make v a (revoked) leaf $\in T$.

The maximum number of subsets that can occur through this method is $2r - 1$, where r is the number of revoked users. We observed that such maximum subsets occur when revoked users are equally scattered throughout the tree. Next, we compare the SD scheme [1] with the self-healing revocation scheme in [18].

5.2.3 Comparing the schemes in [1] and [18]

The broadcast message size for both the SD scheme in [1] and the self-healing revocation scheme in [18] is $O(r)$, where r is the number of revoked users. More precisely, in SD

the maximum number of encryptions of new session keys can be $2r - 1$, and it will also contain a similar number of indices of subsets, so the total elements of $\mathbb{F}_q$ in the message are $4r - 2$, where $\mathbb{F}_q$ is the base finite field. For the scheme in [18], by excluding the parameters for self-healing and considering only those required for key distribution and revocation, there are $4r + 2$ elements of $\mathbb{F}_q$. Hence, the message size in both cases is $O(r)$.

In the SD method, key recovery requires $\log(\log N)$ search operations by a user for his/her index, where N is the number of users in the group. The computation of the subset key requires invoking a pseudo random function G at most $\log N$ times. Finally, a single decryption will recover the new session key. This is in comparison to the second scheme in [18], where each user will have to first compute the polynomial $g_j(x)$, which will require the multiplication of r terms of the form $(x - u_i)$. The evaluation of the resulting polynomial $g_j(x)$ at the user index will require r exponentiations of his/her index ranging from 1 to r, and r multiplications with its coefficients. All operations will take place under modulo q, so modular reduction is also required. Next, the user will have to evaluate the broadcast message at his/her index. If r is a large exponentiation degree then this scheme can be proven to be very expensive. Thus, we claim that key recovery in the scheme in [18] is more expensive than the one in the SD scheme [1].

Keys to store in the SD scheme are on the order of $O(\log^2(N))$, whereas in the other scheme they are proportional to the number of sessions, m. If the network is very large the first approach will require more keys to store, whereas this would be the case with the second scheme if there are more frequent revocations. Thus, the decision of which method provides more savings in terms of storage in a large network may depend on a particular application or environment. It is important to point out that the scheme in [18] will require the group manager to make a prior estimation of the possible sessions in order to pre-compute the masking polynomial for each session. This is not the case in the SD scheme, where key distribution is independent of the number of sessions. In addition and because of this property, the SD scheme does not require a redistribution of any secret information after the number of sessions exceeds the estimation, as is the case in [18].

As a final comparison of the two schemes, we look at their resistance against the collusion of revoked users. The scheme in [18] is secure against the collusion of r revoked users, whereas the SD scheme can revoke any number of users and remain secure against their collusion. Given the discussion in this section, we claim that the SD method proposed in [1] shows better performance than the one in [18]; however it lacks the self-healing feature of the polynomial based scheme.

Next, we describe a method to include a self-healing feature in the SD scheme and discuss optimization techniques that can be used to minimize message size and the required computation in different cases.

5.3 Our approach

To add a self-healing feature for the SD method, we consider three cases.

Case 1 : In this case, we assume that there are no changes to the group membership (no member is added to or removed from the group) between some session s and the following group of t sessions. We can ensure self-healing in the broadcast message for each session by taking the straightforward approach of allowing each broadcast message to contain the respective session key encrypted along with the keys of previous sessions up to session $s+1$. For some subset $S_{i,j}$, the session key K_{s+t} for the session $s+t$ is encrypted with the subset key $L_{i,j}$ along with other keys as:

$$E_{L_{i,j}}(K_{s+t}||K_{s+t-1}||\cdots||K_{s+1})$$

Since all the users in the group are authorized for these sessions, those who did not receive any of the previous session keys can recover that key and the others will discard unwanted keys. The number of encryptions in this case is the same as in the basic SD proposed in [1], however the message size increases by as many the session keys as are made recoverable. In [25, 18], such session keys are sent under masking polynomials and

are concatenated with the current session key in the similar way.

In [1], session keys are randomly chosen and are independent of each other. For optimized self-healing we suggest using a hash chain of session keys for case 1. The server would select a random secret r and compute the hash chain as:

$$h_1 = H(r), h_2 = H(h_1), \cdots, h_t = H(h_{t-1})$$

Keys would be released in reverse order, that is, h_t would be the key for session $s+1$, h_{t-1} for session $s+2$ and so on. Given the current session key, users can compute previous session keys using the one-way hash function H recursively. It is to be noted that the broadcast message will contain only the one session key of the current session and self-healing is possible by computing previous keys using the hash function.

One possible reason for changing session keys without revoking any members could be the use of smaller keys by users with resource-constrained devices in order to improve performance in group communications. Such smaller keys should not be used for a long time and should be refreshed periodically. This can result in different sessions not necessarily revoking any members as in case 1.

Case 2 : In this case, we consider that at any stage after some session s, a group of t sessions occurs in which the members are revoked in some sessions but no new member is added or rejoins. For such cases, we use the same recursive hash chain of the session keys, as in case 1, with some modification. Starting from session s, if $s+i$, for $1 \leq i \leq t$, is the first session in which the first member is revoked; new subsets are constructed by including all previously unauthorized users and those revoked in session $s+i$ in order to protect the new session key h_{t-i+1}. However, the key h_{t-i+2}, for the session $s+i-1$ and all the sessions before it up to $s+1$, should be appended with the current session key in order to ensure self-healing for revoked users who were authorized for the sessions from $s+1$ to $s+i-1$.

For every subsequent removal of members, subsets are made including all previously unauthorized users and the newly revoked members. In general, for all those sessions

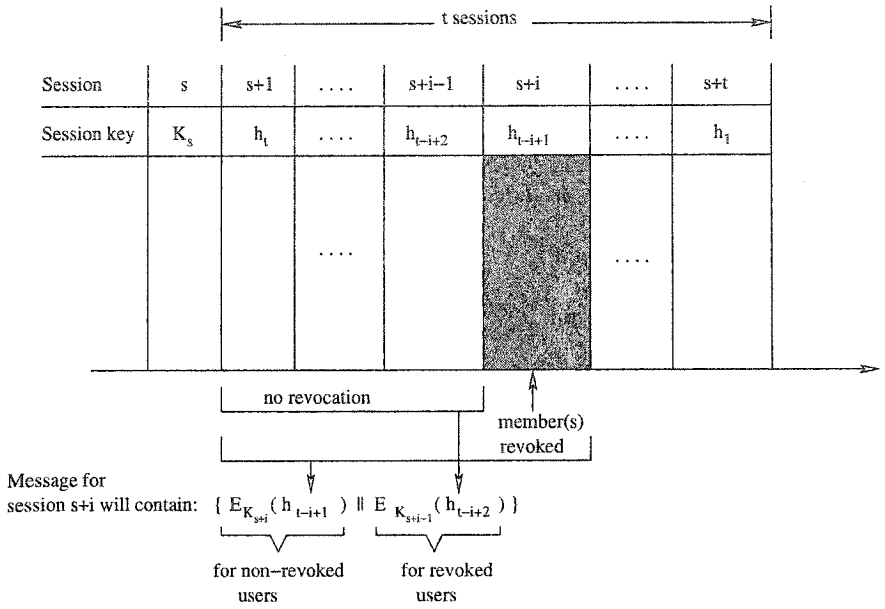


Figure 5.2: Self-healing for revoked and non-revoked users

in which any member is revoked, the key of its preceding session is included in the new session key message. This will ensure self-healing for revoked members to recover those keys for which they were eligible. The optimization we get from recursive hash chains is that those members who remain throughout the period of t sessions, only require one key of current session for self-healing, whereas with the basic approach it requires all the previous keys to be appended to the current session key, as in [25, 18]. And those members who are revoked during this period only require the key for the last session they were eligible to be able to recover previous keys. In other words, the optimization of the case 1 is used on the subgroup level in t sessions. We do not allow addition of new members, or rejoining of the old but revoked members in this case in any session, as such members can extract all session keys of previous sessions for which they were unauthorized.

Case 3 : In this case we consider that after some session s, some members in subsequent sessions can be added to and/or removed from the group. In order to protect previous session keys from unauthorized users, a straightforward approach (or brute force method)

is to make subsets for each session and encrypt its session key with respective subset keys, and append all such encrypted keys with the current session key. This would increase the message size and number of encryptions to be very large. But there are possibilities to optimize it through different methods, some of which are mentioned below.

If same users are revoked in multiple sessions, such session keys can be encrypted once and the SD algorithm will be invoked once. For example, if users u_1, u_2, and u_3 were revoked for some sessions s_1, s_2, and s_3. The subsets based on these revoked users will be made once and three session keys can be encrypted once for each subset. The authorized users also need to decrypt the session keys once. However, we may not find any group of sessions with same revoked members; there can be sessions in which different users were revoked. In order to reduce number of encryptions and decryptions needed in this case, we require that the root maintains a record or a set of revoked users for each session. The root will then find out a subgroup of such revoked sets among which one set is the superset i.e. it contains all the revoked users of other sets in that subgroup. For example, among $s_1 : \{u_1, u_3, u_4\}$, $s_2 : \{u_2, u_5, u_8\}$, $s_3 : \{u_1, u_3\}$ and $s_4 : \{u_1, u_2, u_3, u_4\}$, we find a subgroup of sessions s_1, s_3, and s_4, where s_4 is the superset. The root will then make subsets based on the members in the superset and encrypt all the session keys (of s_1, s_3, and s_4) for those subsets. There can be some users in such superset that might be authorized for any of the sessions in that subgroup, such as in above example user u_2 is authorized for sessions s_1 and s_3, and user u_4 is authorized for session s_3. Such users should be provided those session keys encrypted in their pairwise keys with the root. In this way, we are providing previous session keys to only authorized users and reduce encryption and decryption operations. Recall that in the SD method, the maximum number of subsets can occur is $2r - 1$; considering that, our above optimization will require 10 encryption operations for three session keys, whereas through straightforward approach of repeated application of the SD method to these sessions will require 15 encryptions. Here, as it is expected, the message size increases in our optimization method due to encryption of multiple keys for different subsets. However, the users authorized for all the sessions in that subgroup will still require one decryption operation instead of multiple decryptions through un-optimized method. We require the root to decide either of the

method (repeated application of the SD method or optimization method) based on the affordable message size and the number of encryption and decryption operations required for different messages.

Another method which can provide performance improvements is to limit the number of recoverable session keys to fewer sessions, i.e. using sliding-window technique instead of including all previous sessions. This will be helpful to reduce message size and encryption operations. The minimum size of the window should be greater than or equal to the average number of messages or updates a node can miss consecutively, and the maximum sessions allowed for any user to remain offline i.e. *window size* = *max* [*average packet loss, max. allowed sessions to remain offline*]. In this way upon successful reception of any message a user should be able to recover missing updates.

The periodic relabelling of the nodes can also reduce message size since the administrator does not have to protect keys from the members that are revoked for long time. The decision for relabelling of nodes depends on the complexity of relabelling and the cost associated to keeping track of revoked users. If the cost is high, it may be beneficial to relabel all the nodes which in turn will result reduction in size of subsequent messages.

5.3.1 Mutual healing

A main purpose of the self-healing property is to minimize computational and communication overheads for both the central authority and nodes. But, it is virtually impossible to make nodes completely self-relying without affecting network performance. We thus propose the idea of mutual healing among nodes. It requires nodes to be cooperative with each other and would work in a similar way as the packet relaying is carried out in mobile ad hoc networks. The basic principle behind such mechanism is that a node would cooperate with other node(s) as it might need such cooperation for itself in future. One motivation behind mutual healing is that, if a node has missed an update message, it does not have to wait until the next update message to recover previous session key, instead it can get assistance from its neighboring nodes to recover that key instantly. Such service

is important, for example, in the case of live or real-time transmissions. There are two requirements for mutual healing, first the authentication of requesting node, and second its authorization for the requested session key.

If a node does not receive the broadcast update from the central authority, it will then contact its neighboring nodes to get the missing session key. The neighboring node first needs to authenticate the requesting node. This requires nodes either share a common secret or use public key cryptography to verify authentication. We suggest the use of an identity-based pairwise shared secret proposed in [40], as it requires less communication and is non-interactive if identities are known to the nodes. The challenge-response protocol can be incorporated in the verification process after computing shared secrets. The authentication process will be helpful to identify cooperating or misbehaving nodes, and will allow nodes to provide such service to only authorized users thereby avoiding becoming the target outsider attacks on for their resource consumption.

In order to determine whether to authorize the requesting node for a given session key, we do not require the node that is assisting with the self-healing has to get authorization information from the central authority. Instead, it only needs to forward broadcast message it received from the root. If the requesting node is among the authorized nodes, it would be able to recover the session key(s), otherwise not. This requires nodes to store broadcast messages sent by the root to help other nodes. In order to increase the availability of this service, specifically for mobile environments, nodes should backup complete broadcast messages so that mobile nodes of different subsets should also be able to recover their keys.

5.4 Conclusion

In this chapter, we have proposed the self-healing feature for the Subset Difference (SD) method proposed by D. Naor *et al.*. We have also suggested some optimization techniques that can be used to control the message size and encryption operations needed to ensure a self-healing feature. More work in needed to investigate further optimization by efficient

use of session keys of different type (independent or associated to other keys), and by proper grouping of sessions into subgroups. We have given a comparison of this method to others in the literature, and have shown its possible advantages for large lossy networks, such as ad hoc networks. We have also discussed the notion of mutual healing and some of its requirements; moreover, our future work on mutual healing is focused on its further technical details. One drawback of our optimization approach using recursive hash chains is that if any of such session keys or node itself is compromised, its preceding keys can also be revealed i.e. it trades off backward secrecy at the cost of reducing message size. However, under the condition of case 2, the collusion of any number of users will not allow them to compute keys for the sessions they were not allowed for, and forward secrecy is ensured due to one-way hash operation. A refinement of our optimization or a new approach which guarantees both forward and backward secrecy is also another possible extension of this work.

Chapter 6

Conclusion

6.1 Summary

In our research, we have proposed security solutions for mobile ad hoc networks and provided the security services of confidentiality, authentication, non-repudiation, integrity, key generation and distribution. Our solutions are mainly based on identity-based cryptography which provides easier key management than the conventional public key cryptosystems. We have also presented a solution for self-healing in group key distribution in mobile ad hoc networks as well as in other broadcast environments. We have considered the issues of pairwise and broadcast communication in mobile ad hoc networks and proposed security tools for each.

Using the pairwise key computation proposed in [40], we have proposed a pairwise key management protocol that reduces the key generation and distribution overhead for the Trusted Authority (TA) from $O(N^2)$ to $O(N)$. It also reduces the storage requirement for the nodes, as it does not require nodes to store pairwise keys for all the nodes in the network. Moreover, such pairwise key computation also verifies the authenticity of the users, which required additional communication and involvement of the Certification Authority (CA) in the conventional Diffie-Hellman pairwise key agreement protocol. Thus, in pairwise communication we have proposed a better solution which reduces communication and storage requirements.

For broadcast communication, we have proposed a novel method for computing unique broadcast keys for each node. Given the identities of the broadcast group, such key computation is non-interactive, i.e. does not require any additional communication among the nodes. In order to ensure authentication and non-repudiation in the broadcast protocol, we have proposed an identity-based signature scheme. We have also proposed an identity-based signcryption scheme for authenticated broadcasting that provides the two operations of signature and encryption under one algorithm, and provides the TA with implicit control of the broadcast keys generated by the nodes themselves.

As identity-based cryptosystem suffers from the key escrow problem, we have also proposed a key escrow free solution for both pairwise and broadcast communication protocols. However, in return for increased privacy for the users, it trades off the features of the IBE system, and is only better than conventional PKCS only in not requiring the certificates.

For group key distribution in ad hoc networks, we have studied several proposals and concluded that the Subset Difference (SD) method [1] is the most efficient solutions. Due to the possibility of packet loss in ad hoc networks, we have also proposed self-healing, mutual-healing and optimization techniques for that key distribution method. Most of our proposed solutions have been published in [37, 35, 36] and [38].

6.2 Future work and extensions

We have implemented the software for our proposed solutions using MIRACL cryptographic library. Our future goal is to simulate our solutions with DSR and/or AODV routing protocols for the ad hoc networks using the ns-2 simulator. We have chosen the ns-2 simulator for two reasons: first, it provides the implementation of our desired routing protocols; second, it supports C++ programming other than TCL, which is our requirement since the cryptographic library in use, i.e. MIRACL, is also using C++.

A possible extension to our pairwise-key management protocol could be more precise methods for analyzing which keys are less utilized or might not be used in the near future and can be deleted. This could possibly be learned from information about nodes that have left the network or have moved away from a node, and will therefore interact less often with that node. Another possible extension could be the development of the protocol describing the introduction of a node into a network with no familiar nodes, and the process of developing its buddy-list in such environments using pairwise shared keys to verify the authentication of the claimed identities.

As our broadcast protocol uses public key cryptography in its signature scheme for authentication of broadcast messages, it is computationally expensive. Further research is in progress to reduce the computational requirements for the pairing of points. For example, in [39], P. Barreto *et. al.* have proposed a new pairing technique using non-supersingular curves and have improved the performance of the Tate pairing up to 10 times. It has been suggested that their work could be extended to hyper-elliptic curves, which could result in smaller key sizes and improved performance. Such implementations would reduce the computational overhead of pairing, and would make pairing-based schemes, like ours, more practical. However, any broadcast protocols based on symmetric cryptography providing authentication, will be more appropriate as symmetric cryptography is more efficient than public key cryptography. One such solution proposed in [51] is efficient, but has inappropriate requirements from the perspective of mobile network.

In our solution for self-healing, there is a need for further optimization techniques to control or reduce the message size and encryption or decryption operations. A refinement of our optimization or a new approach which guarantees both forward and backward secrecy is also another possible extension of this work. We have also discussed the notion of mutual healing and some of its requirements; moreover, our future work on mutual healing is focused on its further technical details. Finally, our key distribution mechanism which is not completely distributive, requires such an extension to be more practical for ad hoc network environment.

Appendix A

A.1 Security Problems

A.1.1 The Decisional Diffie-Hellman (BDH) Problem

The Decisional Diffie-Helman (DDH) problem is simple to solve in group $\mathbb{G}_1$, which requires to distinguish between the distributions $< P, aP, bP, abP >$ and $< P, aP, bP, cP >$, where a, b, c are random in $\mathbb{Z}_q^*$ and P is random in $\mathbb{G}_1^*$.

A.1.2 The Computational Diffie-Hellman (BDH) Problem

The Computational Diffie-Hellman (CDH) problem in $\mathbb{G}_1$ is harder than DDH and it requires to find abP, given the points $< P, aP, bP >$.

A.1.3 The Bilinear Diffie-Hellman (BDH) Problem

The BDHP states that, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map, and let P be the generator of $\mathbb{G}_1$, then given $< P, aP, bP, cP >$ for some $a, b, c \in \mathbb{Z}_q^*$, compute $W = e(P, P)^{abc} \in \mathbb{G}_2$. An algorithm $\mathcal{A}$ has advantage ϵ in solving BDHP in $< \mathbb{G}_1, \mathbb{G}_2, e >$ if:

$$Pr[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}] \geq \epsilon$$

A.1.4 Elliptic Curve Discrete Log Problem (ECDLP)

The Elliptic Curve Discrete Log Problem (ECDLP) is defined as, given a point P of order n on an elliptic curve over field $\mathbb{F}_q$ and point aP for some $a \in \mathbb{Z}_n$, it is hard to compute a.

A.2 Security Protocols

A.2.1 Diffie-Hellman key agreement protocol

According to the Diffie-Hellman two party key agreement protocol, two users A and B will generate their respective random integers R_A and R_B. These numbers are drawn from the field $\mathbb{F}_p$ for some prime p. Both users A and B know the parameters p and the generator α. Then, A and B will compute their public keys P_A and P_B as following:

$$P_A = \alpha^{R_A} \ mod \ p,$$

$$P_B = \alpha^{R_B} \ mod \ p$$

Both users exchange their public keys P_A and P_B. Since these values are public and can be sent through unprotected channel. Users A and B will then compute their shared key K_{AB} and K_{BA} respectively as following:

$$K_{AB} = P_B^{R_A} = \alpha^{R_B R_A} \ mod \ p,$$

$$K_{BA} = P_A^{R_B} = \alpha^{R_A R_B} \ mod \ p,$$

The two keys K_{AB} and K_{BA} are equivalent, and thus user A and B can compute their common secret through this protocol known as the Diffie-Hellman key agreement protocol.

Appendix B

Computation of the Weil Pairing

Following examples are taken from [9] to explain computation of Weil pairing.

EXAMPLE 1:

Consider the elliptic curve $y^2 = x^3 + 7x$ defined over $\mathbb{F}_{13}$. The points on $E(\mathbb{F}_{13})$ and their orders are listed in Table 1, where $\#E(\mathbb{F}_{13}) = 18$.

Points	Order	Points	Order
$P_0 = \mathcal{O}$	1	$P_9 = (5, 11)$	6
$P_1 = (0, 0)$	2	$P_{10} = (8, 3)$	6
$P_2 = (2, 3)$	6	$P_{11} = (8, 10)$	6
$P_3 = (2, 10)$	6	$P_{12} = (9, 5)$	3
$P_4 = (3, 3)$	3	$P_{13} = (9, 8)$	3
$P_5 = (3, 10)$	3	$P_{14} = (10, 2)$	3
$P_6 = (4, 1)$	3	$P_{15} = (10, 11)$	3
$P_7 = (4, 12)$	3	$P_{16} = (11, 2)$	6
$P_8 = (5, 2)$	6	$P_{17} = (11, 11)$	6

TABLE 1: $\mathbb{F}_{13}$-rational points on E : $y^2 = x^3 + 7x$

Let $D = 6(P_8) - 6(\mathcal{O})$. The rational function f for it computed as below such that

$div(f) = D$.

$$(P_8) - (\mathcal{O}) = (P_8) - (\mathcal{O}) + div(1)$$

$$2(P_8) - 2(\mathcal{O}) = [(P_8) - (\mathcal{O})] + [(P_8) - (\mathcal{O})]$$

$$= (P_7) - (\mathcal{O}) + div\left[\frac{(-x+y+3)}{(x-4)}\right]$$

$$4(P_8) - 4(\mathcal{O}) = [4(P_8) - 4(\mathcal{O})] + [4(P_8) - 4(\mathcal{O})]$$

$$= (P_6) - (\mathcal{O}) + div\left[\frac{(-x+y+3)^2}{(x-4)^2}\frac{(-5x+y+8)}{(x-4)}\right]$$

$$6(P_8) - 6(\mathcal{O}) = [2(P_8) - 2(\mathcal{O})] + [4(P_8) - 4(\mathcal{O})]$$

$$= div\left[\frac{(-x+y+3)^3}{(x-4)^3}\frac{(-5x+y+8)}{(x-4)}\frac{(x-4)}{1}\right]$$

$$f = \frac{(-x+y+3)^3(-5x+y+8)}{(x-4)^3}$$

EXAMPLE 2:

Considering the same example over the curve $y^2 = x^3 + 7x$. Let $P = P_4 = (3,3)$ and $Q = P_6 = (4,1)$. We shall compute $e_3(P,Q)$.

We first pick random points $T = (8,3), U = (5,2)$ and compute $P + T = (2,10)$, and $Q + U = (5,11)$. Then using the canonical form we compute the rational functions and Weil pairing as:

$$3(P+T) - 3(\mathcal{O}) = (P_1) - (\mathcal{O}) + div\left[\frac{(8x+y)(x+y+1)}{x(x+3)}\right]$$

$$3(T) - 3(\mathcal{O}) = (P_1) - (\mathcal{O}) + div\left[\frac{(11x+y)(8x+y+11)}{x(x+4)}\right]$$

$$3(Q+U)-3(\mathcal{O}) = (P_1)-(\mathcal{O}) + div\left[\frac{(3x+y)(x+y+10)}{x(x+9)}\right]$$

$$3(U)-3(\mathcal{O}) = (P_1)-(\mathcal{O}) + div\left[\frac{(10x+y)(12x+y+3)}{x(x+9)}\right]$$

since $div(f_P) = 3(P+T)-3(T)$, and $div(f_Q) = 3(Q+U)-3(U)$, so by subtracting the second equation from the first equation we get,

$$f_P = \frac{(8x+y)(x+y+1)(x+4)}{(11x+y)(8x+y+11)(x+3)}$$

and by subtracting the last equation from the third equation we get

$$f_Q = \frac{(3x+y)(x+y+10)}{(10x+y)(12x+y+3)}$$

Finally we obtain

$$e_m(P,Q) = \frac{f_P(Q+U)}{f_P(U)} * \frac{f_Q(T)}{f_Q(P+T)}$$

$$= 1 * 9$$

$$= 9$$

Bibliography

[1] D. Naor, M. Naor and J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In *The Proceedings of Advances in Cryptology 2001 - Crypto'01*, Lecture Notes in Computer Science, pages 41–62. Springer-Verlog, 2001.

[2] A. Aziz and W. Diffie. Privacy and Authentication for Wireless Local Area Networks. *IEEE Personal Communications*, 1(1):25–31, 1994.

[3] A. Fiat and M. Naor. Broadcast Encryption. In *The Proceedings of Advances in Cryptology - CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer-Verlog, 1994.

[4] A. Khalili, J. Katz and W.A. Arbaugh. Towards Secure Key Distribution in Truly Ad hoc Networks. In *The Proceedings of the IEEE Workshop on Security and Assurance in Ad hoc Networks in conjunction with the 2003 International Symposium on Applications and the Internet*, Orlando, FL, January 28 2003.

[5] A. Perrig, D. Song and J. D. Tygar. ELK, a New Protocol for Efficient Large-Group Key Distribution. In *The Proceedings of IEEE Symposium on Security and Privacy*, pages 247–262, 2001.

[6] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J.D. Tygar. SPINS: Security Protocols for Sensor Networks. In *The Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking - MobiCom'01*, July 2001.

[7] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *The Proceedings of Advances in Cryptology - Crypto'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlog, 1984.

[8] A. Weimerskirch and G. Thonet. A Distributed Light-Weight Authentication Model for Ad Hoc Networks. In *The Proceedings of the 4th International Conference Seoul on Information Security and Cryptology*, volume 2288 of *Lecture Notes in Computer Science*, pages 341–354, 2002.

[9] A.J. Menezes. *The Handbook of Ad hoc Wireless Networks* . CRC Press, 2003.

[10] B. Dahil, B. Levine, E. Royer and C. Shields. A Secure Routing Protocol for Ad hoc Networks. Technical report, University of Massachusetts, August 2001.

[11] B. Pinkas. Efficient State Updates for Key Management. *Proceedings of the IEEE, Special Issue on Enabling Technologies for Digital Rights Management*, 92(6):910–917, June 2004.

[12] C. Boyd and A. Mathuria. Key Establishment Protocols for Secure Mobile Communications: a selective survey. In *The Proceedings of the 3rd Australian Conference on Information Security and Privacy - ACISP'98*, volume 1438 of *Lecture Notes in Computer Science*, pages 344–355, 1998.

[13] C. Wong and S. Lam. Keystone: A Group Key Management Service. In *The Proceedings of International Conference on Telecommunications - ICT'00*, Acapulco, Mexico, May 2000.

[14] C.K. Wong, M. Gouda and S. Lam. Secure Group Communications Using Key Graphs. In *The Proceedings of ACM SIGCOMM'98*, volume 28(4), Vancouver, BC, Canada, 1998.

[15] D. Boneh and M. Franklin. The Identity-Based Encryption from the Weil Pairing. In *The Proceedings of Advances in Cryptology - Crypto'01*, volume 2139 of *Lecture Notes in Computer Science*, pages 229–231. Springer-Verlog, 2001.

[16] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In *The Proceedings of the Communications of the ACM - CACM'81*, volume 24(2), pages 84–88, February 1981.

[17] D. Halvey and A. Shamir. The LSD Broadcast Encryption Scheme. In *The Proceedings of Advances in Cryptology - Crypto'02*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer-Verlog, 2002.

[18] D. Liu, P. Ning and K. Sun. Efficient Self-Healing Group Key Distribution with Revocation Capability. In *The Proceedings of 10th ACM Conference on Computer and Communications Security - CCS'03*, pages 231–240, Washington D.C., USA, October 2003. ACM Press.

[19] D. Wallner, E. Harder and R. Agee. Key Management for Multicast: Issues and Architectures. Technical report, The Internet Engineering Task Force (IETF), June 1999.

[20] F. Bao, R. Deng, and H. Zhu. Variations of Diffie-Hellman Problem. In *The Proceedings of the International Conference on Information and Communication Security - ICICS'03*, volume 2836 of *Lecture Notes in Computer Science*. Springer-Verlog, October 2003.

[21] H. Tanaka. A Realization Scheme for Identity-Based Cryptosystems. In *The Proceedings of Advances in Cryptology - Crypto'87*, volume 293 of *Lecture Notes in Computer Science*, pages 341–349. Springer-Verlog, 1987.

[22] I. Blake, G. Seroussi and N. Smart., editor. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999.

[23] J. Anzai, N. Matsuzaki, and T. Matsumoto. A quick group key distribution scheme with entity revocation. In *The Proceedings of Advances in Cryptology - ASIACRYPT'99*, volume 1716 of *Lecture Notes in Computer Science*, pages 333–347, Singapore, November 1999. Springer-Verlog.

[24] J-P. Hubaux, L. Buttyan and S. Capkun. The Quest for Security in Mobile Ad Hoc Networks. In *The Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc'01*, 2001.

[25] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean. Self-Healing Key Distribution with Revocation. In *The Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 224–240, 2002.

[26] J.C. Cha and J.H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In *The Proceedings of the International Workshop on Practice and Theory in Public Key Cryptography*, January 2003.

[27] K. Paterson. Cryptography from pairings: a snapshot of current research. Technical report, Royal Holloway, University of London, 2002.

[28] K. Sanzgiri, B. Dahill, B. Levine, C. Shields and E. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. In *The Proceedings of the 10th IEEE International Conference on Network Protocols - ICNP'02*, November 2002.

[29] L. Buttyan and J.-P. Hubaux. Enforcing Service Availability in Mobile Ad Hoc Networks'. In *The Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc'00*, pages 87–96, 2000.

[30] L. Zhou and Z. Hass. Securing Ad hoc Networks. *IEEE Network Magazine*, 13(6):24–30, 1999.

[31] L.S. Charlap and D.P. Robbins. An Elementary Introduction to Elliptic Curves. Technical report, Center for Communications Research,Princeton, 1988.

[32] M. Ilyas. *The Handbook of Ad hoc Wireless Networks* . CRC Press, 2003.

[33] M. Naor and B. Pinkas. Efficient Trace and Revoke Schemes. In *The Proceedings of Financial Cryptography 2000*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20, Anguilla, British West Indies, February 2000. Springer-Verlog.

[34] M. Scott. The Tate Pairing. www.computing.dcu.ie/mike/tate.html.

[35] Muhammad Bohio and Ali Miri. An Authenticated Broadcasting Scheme for Wireless Ad hoc Network. In *The Proceedings of 2nd Annual Conference on Communications Networks Services Research - CNSR'04*, pages 69–74, Fredericton, NB, Canada, May 2004.

[36] Muhammad Bohio and Ali Miri. Authenticated Secure Communications in Mobile Ad hoc Networks. In *The Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering - CCECE'04*, volume 3, pages 1689–1692, Niagara Falls, Canada, May 2004.

[37] Muhammad Bohio and Ali Miri. Efficient Identity-Based Security Schemes for Ad hoc Network Routing Protocols. *Elsevier Science Journal on Ad Hoc Networks - Special Issue on Quality of Service (QoS)*, 2(3):309–317, 2004.

[38] Muhammad Bohio and Ali Miri. Self-healing in Group Key Distribution Using Subset Difference Method. In *The Proceedings of the Workshop on Trustworthy Network Computing (TNC) in conjunction with the IEEE International Conference on the Network Computing and Applications - NCA'04*, page to be published, Cambridge, MA, USA, August-September 2004.

[39] P.S.L.M. Barreto, B. Lynn, M. Scott. On the Selection of Pairing-Friendly Groups. In *The Proceedings of the Selected Areas in Cryptography - SAC'03*, Ottawa, Canada, August 2003.

[40] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing. In *The Proceedings of the 2000 Symposium on Cryptography and Information Security*, Okinawa, Japan, January 2000.

[41] S. Buchegger and J.-Y. Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In *The Proceedings of 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing - EUROMICRO-PDP'02*, pages 403–410, 2002.

[42] S. Jiang, N. Vaidya and W. Zhao. Preventing Traffic Analysis in Packet Radio Network. In *The Proceedings of DARPA Information Survivability Conference and Exposition II - DISCEX'01*, volume 2, pages 158–163, 2001.

[43] S. Lee, S.-M. Hong, H. Yoon and Y. Cho. Accelerating Key Establishment Protocols for Mobile Communication. In *The Proceedings of 4th Australian Conference on Information Security and Privacy - ACISP'99*, volume 1587 of *Lecture Notes in Computer Science*, pages 51–63, April 1999.

[44] S. Marti, T. Giuli, K. Lai and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *The Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking - MobiCom'00*, pages 255–265, Boston, MA, USA, August 2000.

[45] S. Tsuji and T. Itoh. An ID-based Cryptosystem Based on the Discrete Log Problem. *IEEE Journal on Selected Areas in Communication*, 7(4):467–473, 1989.

[46] S.S. Al-Riyami and K.G. Patterson. Certificateless Public Key Cryptography. In *Cryptology ePrint Archive*, Report 2003/126, July 2 2003.

[47] T. Asano. Reducing Storage at Receivers in SD and LSD Broadcast Encryption Schemes. In *The Proceedings of the 4th International Workshop on Information Security Applications - WISE'03*, volume 2908 of *Lecture Notes in Computer Science*, pages 317–332. Springer-Verlog, 2004.

[48] U. Carleson. Optimal Privacy and Authentication on a Portable Communication System. *ACM Operating Systems Review*, 28(3):16–23, 1994.

[49] W. Stallings. *Cryptography and Network Security: Principles and Practice.* Prentice Hall, 2nd edition, 1999.

[50] X. Cheng, X. Huang and D.-Z. Du., editor. *Ad Hoc Wireless Networking,* chapter A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions, pages 319–364. Kluwer Academic Publishers, 2004.

[51] Y.-C. Hu, A. Perrig and D.B. Johnson. Ariadne: A Secure On-demand Routing Protocol for Ad hoc Networks. In *The Proceedings of the 8th Annual International Conference on Mobile Computing and Networking - MobiCom'02*, September 2002.

[52] Y.-C. Hu, A. Perrig and D.B. Johnson. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks. In *The Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[53] Y. Desmedt and J. Quisquater. Public Key Systems Based on the Difficulty of Tampering. In *The Proceedings of the Advances in Cryptology - Crypto'86*, volume 263 of *Lecture Notes in Computer Science*, pages 111–117. Springer-Verlog, 1986.

[54] Y. Yang, X. Li, X. Zhang, and S. Lam. Reliable Group Rekeying: A Performance Analysis. In *The Proceedings of ACM SIGCOMM'01*, pages 27–38, San Diego, CA, USA, August 2001.