Chapter xxx

A Survey of Key Management in Mobile Ad Hoc Networks

Bing Wu
Department of Mathematics and Computer Science
Fayetteville State University
Email: `bwu@uncfsu.edu`

Jie Wu and Mihaela Cardei
Department of Computer Science & Engineering
Florida Atlantic University
Email: `{jie, mihaela}@cse.fau.edu`

Abstract

Security has become a primary concern in mobile ad hoc networks (MANETs). The characteristics of MANETs pose both challenges and opportunities in achieving security goals, such as confidentiality, authentication, integrity, availability, access control, and non-repudiation. Usually, cryptographic techniques are widely used for secure communications in wired and wireless networks. Most cryptographic mechanisms, such as symmetric and asymmetric cryptography, often involve the use of cryptographic keys. In fact, any cryptographic means are ineffective if its key management is weak. Key management is also a central aspect for the security of MANETs. The purpose of key management is to provide secure procedures for handling cryptographic keying materials. The task of key management includes key generation, key distribution, and key maintenance. Key maintenance includes the procedures for key storage, key update, key revocation, key archiving, etc. In MANETs, the computational load and complexity for key management are strongly subject to restriction by the node's available resources and the dynamic nature of network topology. A number of key management schemes have been proposed for the MANETs. In this article, we present a survey of the research work on the key management in MANETs according to recent literature.

Keywords: key management, security, mobile ad hoc network.

1. Introduction

1.1 Mobile Ad Hoc Networks (MANETs)

In areas where there is little communication infrastructure or the existing infrastructure is inconvenient to use, wireless mobile users may still be able to communicate through the

formation of *mobile ad hoc networks* [13]. A mobile ad hoc network, or simply MANET, is a collection of wireless mobile hosts that form a temporary network without the aid of any centralized administration or support. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may be multi-hops from each other.

Possible applications of MANETs include: soldiers relay information for situational awareness on the battlefield; business associates share information during a meeting; attendees use laptop computers to participate in an interactive conference; emergency disaster relief personnel coordinate efforts at sites of fires, hurricanes, or earthquakes.

A routing protocol is necessary in such an environment, since two hosts that wish to communicate may not be able to exchange packets directly. Figure 1 shows a simple example of MANETs. Host *w* is not within the range of host *u*'s wireless transmitter and vice versa. If *u* and *w* wish to exchange packets, they may depend on the services of host *v* to forward packets for them, because *v* is within the overlap between *u* and *w*'s transmission range. Although the number of hops for a host to reach another is likely to be small, the routing problem in a real MANET will still be complicated, due to the inherent non-uniform propagation characteristics of wireless transmissions and the highly dynamic topology of the networks.
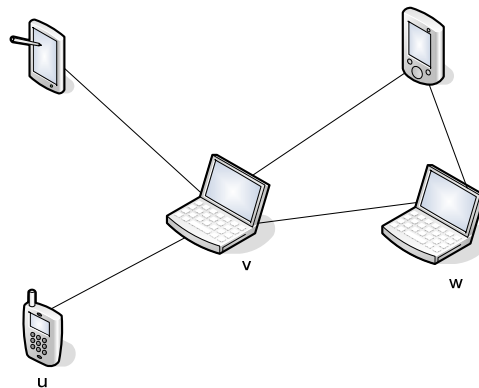

Figure 1   An example of Mobile Ad Hoc Network

1.2 Characteristics of MANETs

A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to an interface with a fixed network. Its nodes are equipped with wireless transmitters/receivers using antennas that may be omni-directional (broadcast), highly directional (point-to-point), or some combination thereof. At a given time, the system can be viewed as a random graph due to the movement of the nodes and their transmitter/receiver coverage patterns, the transmission power levels and the co-channel interference levels [16] [18] [21]. The network topology may change with time as the nodes move or adjust their transmission and reception parameters. Thus, ad hoc networks have several salient characteristics:
- **Dynamic topologies**: The network topology may change randomly and rapidly at unpredictable times, and may consist of both directional and unidirectional links. Nodes

freely roam in the network, join or leave the network at their own will, and fail occasionally.

- **Resource constraints**: The wireless links have significantly lower capacity than wired links. The computation and energy resources of a mobile device are limited.
- **Infrastructure-less**: There is no well-defined infrastructure, or access point, or some other central control point available. Moreover, the wireless medium is accessible by both legitimate nodes and attackers. There is no clear boundary to separate the inside network from the outside world.
- **Limited physical security**: Portable devices are generally small with weak protection. The physical devices could be stolen or compromised.

1.3 Security Challenges Overview

1.3.1 Security attacks

While MANETs can be quickly and inexpensively setup as needed, security is a more critical issue compared to wired networks or other wireless counterparts. Many *passive* and *active* security attacks could be launched from the outside by malicious hosts or from the inside by compromised hosts [4] [7] [20].

- **Passive attacks:** In passive attacks, an intruder captures the data without altering it. The attacker does not modify the data and does not inject additional traffic. The goal of the attacker is to obtain information that is being transmitted, thus violating the message confidentiality. Since the activity of the network is not disrupted, these attacks are difficult to detect. A powerful encryption mechanism can alleviate these attacks, making it difficult to read the transmitted data.
- **Active attacks**: In active attacks, an attacker actively participates in disrupting the normal operation of the network services. An attacker can create an active attack by modifying packets or by introducing false information. Active attacks can be further divided into internal and external attacks: internal attacks are from compromised nodes that were once a legitimate part of the network. Since the adversaries are already part of the network as authorized nodes, they are much more severe [4] and difficult to detect compared to external attacks. External attacks are carried by nodes that are not a legitimate part of the network. Such attacks are often prevented through firewalls or some authentication and encryption mechanisms.

1.3.2 Security goals

Security services include the functionality that is required to provide a secure networking environment. It comprises *authentication*, *access control*, *confidentiality*, *integrity*, *non-repudiation*, and *availability* [9] [12] [15] [22]. Authentication is the ability to verify that a peer entity in an association is the one it claims to be, or can be used for the determination of data origins. Availability ensures the survivability of network service despite denial of service attacks. Confidentiality ensures that certain information is never disclosed to unauthorized entities. Integrity guarantees that a message being transferred is not corrupted. Non-repudiation ensures that the origin of a message cannot deny having sent the message. Access control is the ability to limit and control access to devices and/or applications via communication links. The main security services can be summarized as follows:

- **Authentication**: The function of the authentication service is to verify a user's identity and to assure the recipient that the message is from the source that it claims to be from. First, at the time of communication initiation, the service assures that the two parties are authentic, that each is the entity it claims to be. Second, the service must assure that a third party does not interfere by impersonating one of the two legitimate parties for the purpose of authorized transmission and reception.
- **Confidentiality**: Confidentiality ensures that the data/information transmitted over the network is not disclosed to unauthorized users. Confidentiality can be achieved by using different encryption techniques such that only legitimate users can analyze and understand the transmission.
- **Integrity:** The function of integrity control is to assure that the data is received exactly as sent by an authorized party. That is, the data received contains no modification, insertion, deletion, or replay.
- **Access control**: This service limits and controls the access of a resource such as a host system or application. To achieve this, a user trying to gain access to the resource is first identified (authenticated) and then the corresponding access rights are granted.
- **Non-Repudiation:** This is related to the fact that if an entity sends a message, the entity cannot deny that it sent that message. If an entity gives a signature to the message, the entity cannot later deny that message. In public key cryptography, a node A signs the message using its private key. All other nodes can verify the signed message by using A's public key, and A cannot deny the message with its signature.
- **Availability:** This involves making network services or resources available to the legitimate users. It ensures the survivability of the network despite malicious incidences.

1.3.3 Security mechanisms

Cryptography is an important and powerful tool for secure communications. It transforms readable data (plaintext) into meaningless data (ciphertext). Cryptography has two dominant categories, namely symmetric-key (secret-key) and asymmetric-key (public-key) approaches [1]. In symmetric-key cryptography, the same key is used to encrypt and decrypt the messages, while in the asymmetric-key approach, different keys are used to convert and recover the information. Although the asymmetric cryptography approaches are versatile (can be used for authentication, integrity, and privacy) and are simpler for key distribution than the symmetric approaches, symmetric-key algorithms are generally more computation-efficient than the asymmetric cryptographic algorithms [2]. There are varieties of symmetric and asymmetric algorithms available, like DES, AES, IDEA, RSA, and EIGamal [25]. Threshold cryptography is another cryptographic technique that is quite different from the above two approaches. In Shamir's (k, n) secret sharing scheme, secret information is split into n pieces according to a random polynomial; meanwhile, the secret could be recovered by combining any threshold k pieces based on Lagrange interpolation. These cryptographic algorithms are the security primitives that are widely used in wired and wireless networks. They can also be used in MANETs and help to achieve the security in its unique network settings.

1.3.4 Key management

As in the above description, cryptography is a powerful tool in achieving security. However, most cryptosystems rely on the underlying secure, robust, and efficient key management subsystem. In fact, any cryptographic means is ineffective if the key management is weak. Key

management is a central part of the security of MANETs. In MANETs, the computational load and complexity for key management are strongly subject to restriction by the node's available resources and the dynamic nature of network topology. Some asymmetric and symmetric (including group key) key management schemes have been proposed to adapt to the environment of MANETs. Key management deals with key generation, key storage, distribution, updating, revocation, deleting, archiving, and using keying materials in accordance with security policies. In the article, we present a comprehensive survey of research work on key management in MANETs based on recent literature. This article is organized as follows: Section 1 gives an introduction of MANETs. Section 2 discusses key management and trust models in wired networks and MANETs. Section 3 presents the asymmetric key management schemes in MANETs. Section 4 presents the symmetric key management schemes in MANETs. The group key management schemes are shown in Section 5. In Section 6, we conclude the article and discuss possible future work.

2. Key Management in MANETs

Key management is a basic part of any secure communication. Most cryptosystems rely on some underlying secure, robust, and efficient key management system. Secure network communications normally involve a key distribution procedure between communication parties, in which the key may be transmitted through insecure channels. A framework of trust relationships needs to be built for authentication of key ownership in the key distribution procedure. While some frameworks are based on a centralized trusted third party (TTP), others could be fully distributed. For example, a certificate authority (CA) is the TTP in asymmetric cryptosystems; a key distribution center (KDC) is the TTP in the symmetric system, while in PGP no TTP is assumed. According to recent literature, the centralized approach is regarded as inappropriate for MANETs because of the dynamic environment and the transient relationship among mobile nodes. Most researchers prefer the decentralized trust model for the MANETs. Several decentralized solutions have been proposed in recent papers with different implementations, such as how CA's responsibility is distributed to all nodes, a subset of nodes, or even no such virtual CA at all in the web-of-trust model.

2.1 Fundamentals of Key Management

Cryptographic algorithms are security primitives that are widely used for the purposes of authentication, confidentiality, integrity, and non-repudiation. Most cryptographic systems require an underlying secure, robust, and efficient key management system. Key management is in the central part of any secure communication and is the weakest point of system security and the protocol design.

A key is a piece of input information for cryptographic algorithms. If the key was released, the encrypted information would be disclosed. The secrecy of the symmetric key and private key must always be assured locally. The Key Encryption Key (KEK) approach [8] could be used at local hosts to protect the secrecy of keys. To break the circle (use key to encrypt the data, and use key to encrypt key) some non-cryptographic approaches need to be used, e.g. smart card, or biometric identity, such as fingerprint, etc.

Key distribution and key agreement over an insecure channel are at high risk and suffer from potential attacks. In the traditional digital envelop approach, a session key is generated at one side and is encrypted by the public-key algorithm. Then it is delivered and recovered at the other end. In the Diffie-Hellman (DH) scheme [8], the communication parties at both sides exchange some public information and generate a session key on both ends. Several enhanced DH schemes have been invented to counter man-in-the-middle attacks. In addition, a multi-way challenge response protocol, such as Needham-Schroeder [19], can also be used. Kerberos [19], which is based on a variant of Needham-Schroeder, is an authentication protocol used in many real systems including Microsoft Windows. However, in MANETs, the lack of a central control facility, the limited computing resources, dynamic network topology, and the difficulty of network synchronization contribute to the complexity of key management protocols.

Key integrity and ownership should be protected from advanced key attacks. Digital signature, hash function, and hash function based message authentication code (HMAC) [25] are techniques used for data authentication and/or integrity purposes. Similarly, the public key is protected by the public-key certificate, in which a trusted entity called the certification authority (CA) in PKI vouches for the binding of the public key with the owner's identity. In systems lacking a TTP, the public-key certificate is vouched for by peer nodes in a distributed manner, such as pretty good privacy (PGP) [8]. In some distributed approaches, the system secret is distributed to a subset or all of the network hosts based on threshold cryptography. Obviously, a certificate cannot prove whether an entity is "good" or "bad", but can prove ownership of a key. Mainly it is for key authentication.

A cryptographic key could be compromised or disclosed after a certain period of usage. Since the key should no longer be usable after its disclosure, some mechanism is required to enforce this rule. In PKI, this can be done implicitly or explicitly. The certificate contains the lifetime of validity - it is not useful after expiration. However, in some cases, the private key could be disclosed during the valid period, in which case the CA needs to revoke a certificate explicitly and notify the network by posting it onto the certificate revocation list (CRL) to prevent its usage.

Key management for large dynamic groups is a difficult problem because of scalability and security. Each time a new member is added or an old member is evicted from the group, the group key must be changed to ensure backward and forward security. Backward security means that new members cannot determine any past group key and discover the previous group communication messages. Forward security means that evicted members cannot determine any future group key and discover the subsequent group communication information. The group key management should also be able to resist against colluded members.

2.2 Trust Models

2.2.1 Centralized trust model

For the centralized trust model, there is a well-trusted entity called TTP [4] [23] [25]. A TTP is an entity trusted by all users in the system, and it is often used to provide key management services. Depending on the nature of their involvement, TTPs can be classified into three categories: namely inline, online, or offline. See Figure 2 for an illustration. An inline TTP participates actively in between the communication path of two users while online TTP

participates actively but only for management purposes, the two parties communicate with each other directly. An offline TTP communicates with users prior to the setting up of communication links and remains offline during network operation.
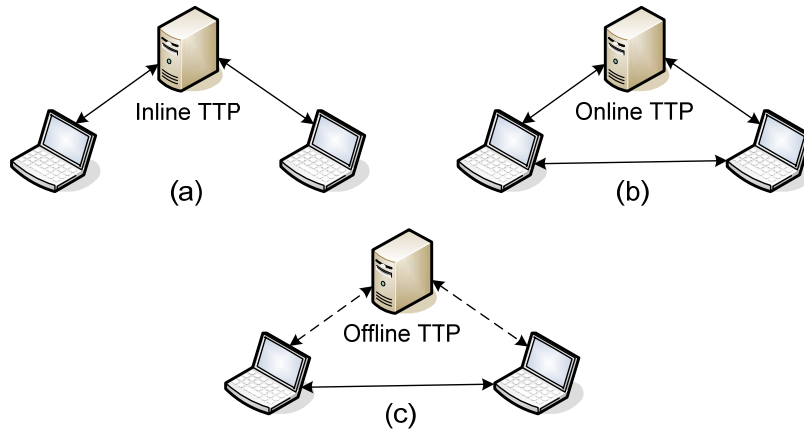

Figure 2 Categories of trust third parties

2.2.1.1 TTPs in symmetric key management systems

TTPs have been implemented in both symmetric and asymmetric key management systems. Key Distribution Centers (KDC) and Key Translation Centers (KTC) [14] are TTPs in symmetric cryptographic key management systems and the certification authority (CA) is the TTP in public-key management systems. KDC and KTC simplify the symmetric key management since each user does not have to share a secret key with every other user; instead, it only needs to share one key with the TTP. This reduces the total number of keys that need to be managed from $\frac{n(n-1)}{2}$ to n, where n is the total number of users. Figure 3 illustrates the protocols by implementing KDC or KTC.
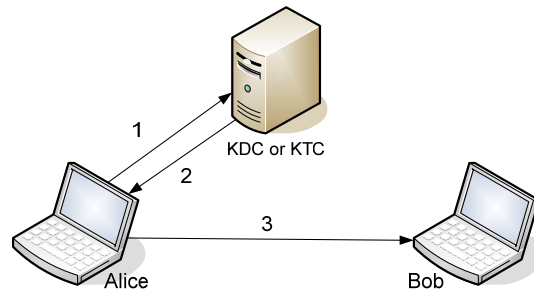

Figure 3 Establishment of session key using KDC or KTC

1 Alice requests to share a secret key with Bob. If the TTP is KDC, it generates a key to use. Otherwise, Alice provides it. The message is encrypted using the secret key shared between Alice and the TTP.

2 The TTP encrypts the session key with the key it shares with Bob and returns it to Alice.

3 Alice sends the encrypted session key to Bob, who can decrypt it and thereafter use it to communicate securely with Alice.

2.2.1.2 Public key infrastructure (PKI)

The use of public key cryptography requires the authenticity of public keys. Otherwise, it is easy to forge or spoof someone's public key. Some trusted framework must be present to verify the ownership of a public key. A straightforward solution is to have any two users that wish to communicate exchange their public keys in an authenticated manner. It would require the initial distribution of $n(n-1)$ public keys. Obviously, this solution is not scalable for a large network and has the same problems we discussed in the symmetric key management system. However, by having a trusted third party issue certificates to each of the users, every user only needs to hold the public key of the TTP, which significantly simplifies the authentication process for users' public keys. Actually, there are two dominating trust models in PKI, namely, centralized and web-of-trust trust models [4] [10]. For network scalability, the centralized trust model could be a hierarchical trust structure instead of a single CA entity. Multiple CA roots could be necessary for a large network, such as the Internet. We will discuss the fully distributed or web-of-trust model later.

A PKI provides the mechanisms needed to manage certificates, and normally consists of the components illustrated in Figure 4.
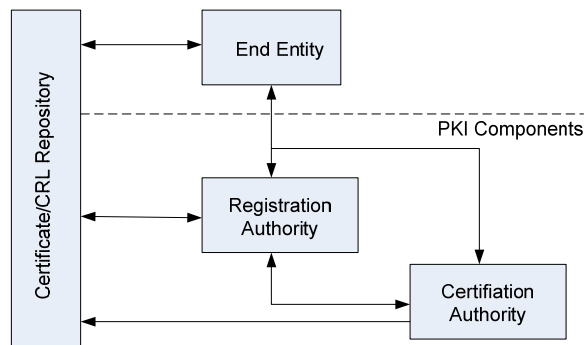

Figure 4 Components of a PKI

In this diagram, the certification authority (CA) is the component responsible for issuing and revoking certificates, while the registration authority (RA) is responsible for establishing the identity of the subject of a certificate and the mapping between the subject and its public key. The RA and CA can be implemented as one component; therefore, RA is an optional component. PKI components provide basic services, such as registration, initialization, certification, key update, revocation, key recovery, cross-certification, etc.

2.2.2 Web-of-trust model

The web-of-trust model is also called certificate chaining. PGP [19] is an example built on this trust model. In the web-of-trust model there is no TTP that is well-trusted by all network nodes. Instead, peer nodes can issue certificates to each other and populate the certificate graph. Certificates can be authenticated through certificate chaining. Compared with the centralized trust model, the web-of-trust model does not require a heavy infrastructure or complex bootstrapping procedures, and every node plays an identical role and shares the same responsibility. Although the web-of-trust model has the above advantages, it has two major limitations. First, a certificate graph may not populate enough to provide certificate chains for a

given pair of nodes, so it is difficult to predict whether any given authentication request can be fulfilled. Second, without relying on a TTP, any trust relationship relies on the goodwill and the correct behaviors of all participants. Obviously, that cannot always be assumed. However, since there is no clear way to tell if a certificate chain includes any misbehaving nodes, the overall confidence for the certificate is relatively low.

2.2.3 Decentralized trust model

In MANETs, a framework for key management built on a fully centralized mode is not feasible, not only because of the difficulty of maintaining such a globally trusted entity but also because the central entity could become a hot spot of attacks. Thus, this network suffers from a security bottleneck. Meanwhile a completely distributed model may not be acceptable because there is no well-trusted security anchor available in the whole system. One feasible solution is to distribute the central trust to multiple (or entire network) entities based on a secret sharing scheme. In the decentralized public key management scheme, the system public key is distributed to the entire network, while the system private key is split to multiple pieces and distributed to a subset (or all) of the nodes. The subset of group nodes creates a view of a CA and functions as a CA in combination.

2.2.4 Hybrid trust model

This scheme takes advantage of the positive aspects of two different trust systems. The basic idea is to incorporate a TTP into the certificate graph. Here, the TTP is a virtual CA node that represents all nodes that comprise the virtual CA. Some authentication metrics, such as confidence value, are introduced in order to "glue" two trust systems [10]. While this model is theoretically sound, it is difficult to "glue" two different trust systems since there is no clear way to assign a value of confidence level.

2.3 Overview of Key Management Schemes in MANETs

2.3.1 Asymmetric key management schemes

Recently, research papers have proposed different key management schemes for MANETs. Most of them are based on public-key cryptography. The basic idea is to distribute the CA's functionality to multiple nodes. Zhou and Hass [3] presented a secure key management scheme by employing *(t, n)* threshold cryptography. The system can tolerate t-1 compromised servers. Luo, Kong, and Zerfos proposed a localized key management scheme [6] [44] in which all nodes are servers and the certificate service can be performed locally by a threshold number of neighboring nodes. Yi, Naldurg, and Kravets [5] put forward a similar scheme. The difference is that the certificate service is distributed to a subset of nodes, which are physically more secure and powerful than the others. Wu and Wu [15] also introduced a scheme that is similar to Yi, in which server nodes form a mesh structure and a ticket scheme is used for efficiency. Capkun, Buttyan, and Hubaux [28] considered a fully distributed scheme that is based on the same idea of PGP. Yi and Kravets [10] provided a composite trust model. Their idea is to take advantage of the positive aspects of both the central and fully distributed trust models.

2.3.2 Symmetric key management schemes

There are research papers that are based on the symmetric-key cryptography for securing MANETs. For instance, some symmetric key management schemes are proposed for sensor nodes that are assumed incapable of performing costly asymmetric cryptographic computations. Pairwise keys can be preloaded into nodes, or based on the random key distribution in which a set of keys is preloaded. Chan [40] introduced a distributed symmetric key distribution scheme for MANETs. The basic idea is that each node is preloaded with a set of keys from a large key pool [37] [38]. The key pattern should satisfy the property that any subset of nodes can find at least one common key, and the common key should not be covered by a collusion of a certain number of other nodes outside the subset. Chan and Perrig [39] introduced a symmetric key agreement scheme for the sensor nodes. The basic idea of their approach is that each node shares a unique key with a set of nodes vertically and horizontally (in case of 2-Dimension). Therefore, any pair of nodes can rely on at least one intermediate node to establish the common key.

2.3.3 Group key management schemes

Collaborative and group-oriented applications in MANETs are going to be active research areas. Group key management is one of the basic building blocks in securing group communications. However, key management for large dynamic groups is a difficult problem because of scalability and security [32]. For instance, each time a new member is added or an old member is evicted from the group, the group key must be changed to ensure backward and forward security.

3. Asymmetric Key Management Schemes in MANETs

3.1 Secure Routing Protocol (SRP)

SRP is a decentralized public key management proposed by Zhou and Hass [3] by employing *(t, n)* threshold cryptography [41] [42] [43] in their research paper called "Securing Ad Hoc Networks". In the system, there are *n* servers, which are responsible for public-key certificate service. Therefore, the system can tolerate *t-1* compromised servers. Servers can proactively refresh the secret shares using the proactive secret sharing (PSS) [24] techniques or by adjusting the configuration structure based on share redistribution techniques to handle compromised servers or system failure. Since the new shares are independent of the old ones, mobile adversaries would have to compromise a threshold number of servers in a very short time frame, which obviously increases the difficulty for the success of adversaries. The system configuration of this scheme is illustrated in the Figure 5. The system public key K is distributed to all nodes in the network, whereas the private key S is split to *n* shares $s_1, s_2, s_3, \ldots, s_n$, one share for each server according to a random polynomial function.
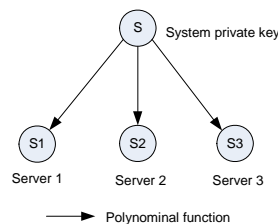

Figure 5 Illustration of SRP scheme

10

In this scheme, the system model is that the *n* servers are special nodes, each having its own public/private key pair and the public key of every node in the network. This is a critical issue in a large network. However, this scheme does not describe how a node can contact *t* servers securely and efficiently in case the servers are scattered in a large area. A share-refreshing scheme is proposed to counter mobile adversaries. The update of secret shares does not change the system public/private key pairs. Therefore, nodes in the network can still use the same system public key to verify a signed certificate so the share refreshing is transparent to all nodes. However, how to distribute these updated subshares to all nodes securely and efficiently in the network is not addressed.

3.2 Ubiquitous and Robust Access Control (URSA)

URSA is a localized key management scheme proposed by Luo, Kong, and Zerfos [6] [44] in their paper "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks". The URSA protocol is also based on threshold cryptography as in SRP [3]. The difference between URSA and SRP is that in URSA, all nodes are servers and are capable of producing a partial certificate, while in SRP only server nodes can produce certificates. Thus, certificate services are distributed to all nodes in the network. URSA also proposed a distributed self-initialization phase that allows a newly joined node to obtain secret shares by contacting a coalition of *k* neighboring nodes without requiring the existence of an online secret share dealer. The basic idea is to extend the PSS technique by shuffling the partial shares instead of shuffling the secret sharing polynomials. The purpose of this shuffling process is to prevent deducing the original secret share from a resulting share.

In URSA, every node should periodically update its certificate. To update its certificate, a node must contact its 1-hop neighbors, and request partial certificates from a collection of threshold *k* number of nodes. It can combine partial certificates into a legitimistic certificate. This will introduce either communication delay or cause search failure. It could potentially utilize services from 2-hop neighboring nodes.

The advantage of this scheme is efficiency and secrecy of local communications, as well as system availability since the CA's functionality is distributed to all network nodes. On the other hand, it reduces system security, especially when nodes are not well-protected because an attack can easily locate a secret holder without much searching and identifying effort. One problem is that in a sparse network where a node has a small number of neighbors, the threshold *k* is much larger than the network degree *d* and a node that wants to have its certificate updated needs to move around in order to find enough partial certificate "producer". The second critical issue is the convergence in the share-updating phase. Another critical issue is that too much off-line configuration is required before accessing the networks.

3.3 Mobile Certificate Authority (MOCA)

MOCA is a decentralized key management scheme proposed by Yi, Naldurg, and Kravets [5] in the paper "Key management for heterogeneous ad hoc wireless networks". In this approach, a certificate service is distributed to Mobile Certificate Authority (MOCA) nodes. MOCA nodes are chosen based on heterogeneity if the nodes are physically more secure and computationally more powerful. In cases where nodes are equally equipped, they are selected randomly from the

network. The trust model of this scheme is a decentralized model since the functionality of CA is distributed to a subset of nodes. A service-requesting node can locate $k + \alpha$ MOCA nodes randomly, or based on the shortest path, or according to the freshest path in its route cache. However, the critical question is how nodes can discover those paths securely since most secure routing protocols are based on the establishment of a key service in advance.

3.4 Self-organized Key Management

Capkun, Buttyan, and Hubaux [19] considered a fully distributed key management scheme in their paper "Self-organized public key management for mobile ad hoc networks". This scheme is based on the web-of-trust model that is similar to PGP [8]. The basic idea is that each user acts as its own authority and issues public key certificates to other users. A user needs to maintain two local certificate repositories, one is called the non-updated certificate repository and the other one is called the updated certificate repository. The reason a node maintains a non-updated certificate repository is to provide a better estimate of the certificate graph. Key authentication is performed via chains of public key certificates that are obtained from other nodes through certificate exchanging, and are stored in local repositories.
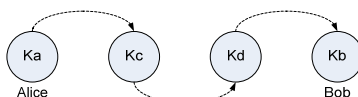

Figure 6 An example of certificate chain

The fully distributed, self-organized certificate chaining has the advantage of configuration flexibility and it does not require any bootstrapping of the system. However, this certificate chaining requires a certain period to populate the certificate graph. This procedure completely depends on the individual node's behavior and mobility. One the other hand, this fully self-organized scheme lacks any trusted security anchor in the trust structure that may limit its usage for applications where high security assurance is demanded. In addition, many certificates need to be generated and every node should collect and maintain an up-to-date certificate repository. The certificate graph, which is used to model this web of trust relationship, may not be strongly connected, especially in the mobile ad hoc scenario. In that case, nodes within one component may not be able to communicate with nodes in different components. Certificate conflicting is another potential problem in this scheme.

3.5 Composite Key Management

Recently, Yi, and Kravets [10] provided a composite key management scheme in their paper titled "Composite key management for ad hoc networks". In their scheme, they combine the centralized trust and the fully distributed certificate chaining trust models. This scheme takes advantage of the positive aspects of two different trust systems. The basic idea is to incorporate a TTP into the certificate graph. Here, the TTP is a virtual CA node that represents all nodes that comprise the virtual CA. Some authentication metrics, such as confidence value, are introduced in order to "glue" two trusted systems. A node certified by a CA is trusted with a higher confidence level. However, proper assignment of confidence values is challenging. An example of composite key management model is shown in Figure 7.
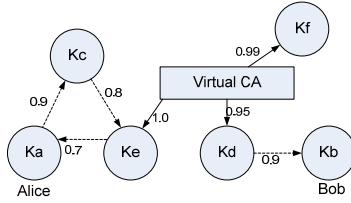
Figure 7 An example of composite key management scheme

3.6 Secure and Efficient Key Management (SEKM)

SEKM is a decentralized key management scheme proposed by Wu and Wu [15] [17] in the paper "secure and efficient key management in mobile ad hoc networks". It is based on the decentralized virtual CA trust model. While all decentralized key management schemes are quite similar, that is, the functionality of CA is distributed to a set of nodes based on the techniques of threshold cryptography. No schemes except SEKM present detailed, efficient, and secure procedures for communications and cooperation between secret shareholders that have more responsibilities. In SEKM, all servers that have a partial system private key are to connect and form a server group. The structure of the server group is a mesh structure as shown in Figure 8. Periodic beacons are used to maintain the connection of the group so servers can efficiently coordinate with each other for share updates and certificate service. The problem of SEKM is that for a large network with highly dynamic mobility, maintaining the structure server group could be costly.
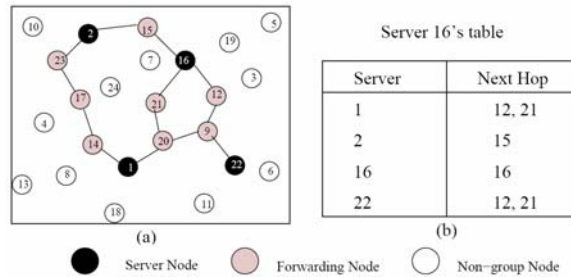

Figure 8 Server group structure in SEKM

4. Symmetric Key Management Schemes in MANETs

4.1 Distributed Key Pre-distribution Scheme (DKPS)

DKPS is a distributed symmetric key management scheme proposed by Chan [40] in the paper titled "Distributed symmetric key management for mobile ad hoc networks". It is aimed at the network settings where mobile nodes are not assumed to be capable of performing computationally intensive public key algorithms and the TTP is not available. The basic idea of the DKPS scheme is that each node randomly selects a set of keys in a way that satisfies the probability property of cover-free family (CFF). Any pair of nodes can invoke the secure shared

13

key discovery procedure (SSD). The theory behind the SSD is the additive and scalar multiplicative homomorphism of the encryption algorithm as well as the property of non-trivial zero encryption. To discover the common secret key, one side of the two parties can form a polynomial and send the encrypted polynomial to the other side. The coefficients of the polynomial are encrypted with the sender's secret key. The other side will send back the encrypted polynomial multiplied by a random value. Because of the homomorphism and non-trivial zero encryption properties, either side can only discover the common secret key, without disclosing the other non-common keys.

4.2 Peer Intermediaries for Key Establishment (PIKE)

PIKE is another symmetric key management scheme proposed by Chan and Perrig [39] in their paper entitled "PIKE: Peer intermediaries for key establishment in sensor networks". It is a random key pre-distribution scheme. The basic idea of PIKE is to use sensor nodes as trusted intermediaries to establish shared keys. Each node shares a unique secret key with a set of nodes. In case of 2-Dimension, a node shares a unique secret with each of the $O(\sqrt{n})$ nodes in the horizontal and vertical dimensions. Therefore, any pair of nodes can have a common secret with at least one intermediate node. This key pre-distribution scheme can be extended to three or more dimensions. Figure 9 shows the basic idea of the PIKE scheme. Dark lines are the nodes that each one shares a unique key with node A, and light lines are nodes that each one shares a unique key with node B. There are six nodes that each one shares a unique key with node A and node B.
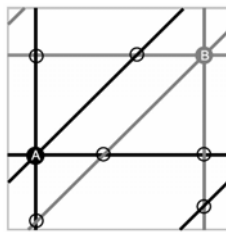

Figure 9 Illustration of PIKE scheme

5. Group Key Management Approaches

The messages are protected by encryption using the chosen key, which in the context of group communication is called the group key. Only those who know the current group key are able to recover the original message. Group key establishment means that multiple parties want to create a common secret to be used in the secure exchange of information. Two people who did not previously share a common secret can create one common secret with a DH key exchange protocol. The 2-party DH protocol can be extended to a generalized version of *n*-party DH key-exchange model. Research efforts have been put into the design of group key agreement protocols to achieve better scalability, efficiency, and storage saving, such as the introduction of a tree structure and hash function. Furthermore, the group key management also needs to address

the security issue related to membership changes. The modification of membership could require the group key to be refreshed (e.g., periodic re-key). The change of group keys when old members leave or new members join ensures backward and forward security. Therefore, a group key scheme must provide a scalable and efficient mechanism to re-key the group.

Group key management protocols can be roughly classified into three categories, namely, centralized, decentralized, and distributed [32]. In centralized group key protocols, a single entity is employed to control the whole group and is responsible for re-keying and distributing group keys to group members. In the decentralized approaches, a set of group managers is responsible for managing the group instead of a single entity. In the distributed method, group members themselves contribute to the formation of group keys and are equally responsible for the re-keying and distribution of group keys. Recently, collaborative and group-oriented applications in MANETs have been an active research area. Obviously, group key management is a central building block in securing group communications in MANETs. However, group key management for large and dynamic groups in MANETs is a difficult problem because of the requirement of scalability and security under the restrictions of nodes' available resources and unpredictable mobility.

The literature presents several approaches to group key management. In this section, we give an overview of those protocols. Most of the following group key protocols are designed for the infrastructure networks. However, with the proper extension, some of them could be utilized and adapted to the MANET environment, or could serve as a hint for the design of MANET-specific group key management protocols. For instance, GDH (Section 5.4) and LKH (Section 5.1) have been extended into the MANETs. [45] proposed a simple and efficient group key management scheme, called SEGK, for MANETs. The basic idea of SEGK is that a physical multicast tree is formed in MANETs for efficiency. Group members take turns acting as group coordinator to compute and distribute intermediate key materials to group members. The keying materials are delivered through the tree links. The coordinator is also responsible to maintain the connection of the multicast group. All group members can compute the group key locally in a distributed manner.

5.1 Logical Key Hierarchy (LKH)

LKH is a centralized group key management scheme proposed by Wallner, Harder and Agee [34] [35]. It is based on the tree structure with each user (group participant) corresponding to a leaf and the group initiator as the root node. The tree structure will significantly reduce the number of broadcast messages and storage space for both the group controller and group members. The operation of this scheme is outlined below.

Each leaf node shares a pairwise key with the root node as well as a set of intermediate keys from it to the root. So, for a balanced binary tree, each group member stores at most $d+1$ keys, where $d = \log_2 n$, is the height of the tree, and n is the total number of group members. See Figure 10: U_5 stores $k5$, $k56$, $k58$, and $k0$.
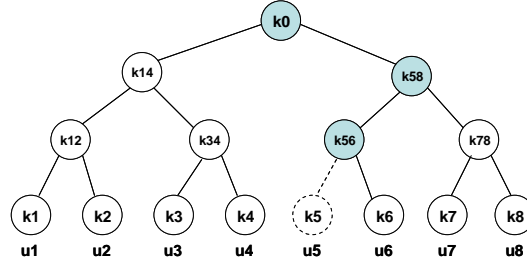
Figure 10 A sample tree structure of LKH

When a member joins the group, the re-key procedure will be started. A re-key message is generated containing the new set of keys encrypted with its respective node's children key. Figure 10 shows keys that are affected. The new member U_5 receives a secret key $k5$ and attaches the intermediate node k56 logically. The keys $k56$, $k58$ and $k0$, which are in the path from $k5$ to $k0$ need to be refreshed. New keys, $k'56$, $k'58$, and $k'0$, are generated as illustrated in Figure 11. These keys are encrypted with their respective node's children's key, e.g., one instance of $k'56$ is encrypted by $k5$, and the other copy is encrypted by $k6$ (see Figure 11). Removing a member follows a similar procedure. For instance, when member U6 leaves the group, $k56$, $k58$, and $k0$ should be changed and the new set of keys $k'56$, $k'58$, and $k'0$ are encrypted with their respective children's key. See Figure 12 for an illustration of a member leave.
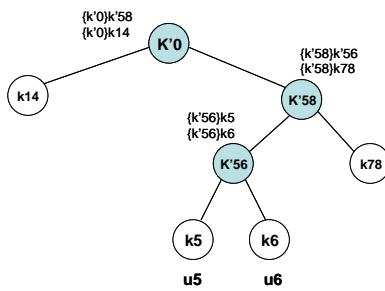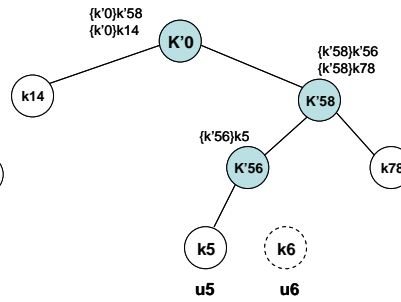

Figure 11: Illustration of joining member U5   Figure 12: Illustration of leaving member U6

5.2 One-Way Function Trees (OFT)

OFT is another centralized group key management scheme proposed by Sherman and McGrew [27]. It is based on the tree structure that is similar to the above LKH scheme. However, all keys in the OFT scheme are functionally related according to a one-way hash function. The idea is that the keys held by a node's children are blinded using a one-way hash function and then combined together using a mixing function, such as a bitwise exclusive-or operation. Each group user receives blind keys from its sibling set as well as the blind key of its own sibling. Based on collected blinded keys, the group users can deduce each key of its ancestor set. See Figure 13 for an illustration. $k6$ is the key of U5's sibling. $k56$, $k58$, and $k0$ are the keys of U5's ancestor set. $k78$ and $k14$ are the keys of U5's sibling set.

A group user still needs to store $d+1$ keys, where $d = \log_2 n$ is the height of the tree, and n is the total number of group members. The scheme has the same complexity as the LKH scheme for a

16

balanced tree structure, but in the re-keying process, the size of keying materials reduces from $2 \cdot \log_2 n$ to $\log_2 n$.
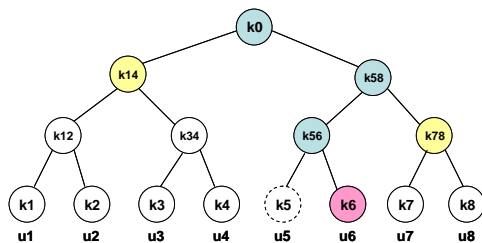

Figure 13 A sample tree structure of OFT

The message size reduction is achieved because in the OFT scheme, the blinded key changed in a node is encrypted only with the key of its sibling node while in LKH scheme the new key must be encrypted with its two children's keys, see Figure 14.
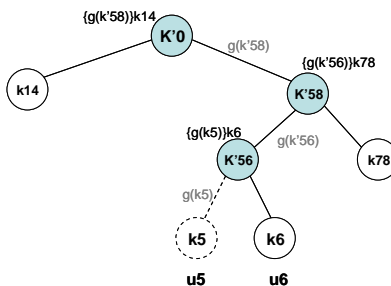

Figure 14: Illustration of join member U5 in OFT

5.3 Tree-Based Group Diffie-Hellman (TGDH)

TGDH is a group key management scheme proposed by Kim, Perrig, and Tsudik [29] [30]. It is a tree-based group DH scheme. The basic idea is to combine the efficiency of the tree structure with the contributory feature of DH.

The basic operation of this scheme is as the follows. Each group member contributes its (equal) share to the group key, which is computed as a function of all the shares of current group members. As the group grows, new members' shares are factored into the group key but old members' shares remain unchanged. As the group shrinks, departing members' shares are removed from the new key and at least one remaining member changes its share. All protocol messages are signed by the sender using RSA.

In TGDH, a sponsor takes a special role that can involve computing keys and broadcasting the blinded keys to the group during events of member join, leave, partition, and merge. Any member in the group can take on this responsibility. Figure 15 illustrates the operation of member join. When M_4 joins the group, sponsor M_3 will rename node <1,1> to <2, 2>; generate a new intermediate node <1,1> and new member node <2,3>; promote <1, 1> as the parent node of <2,2> and <2,3>. Sponsor M_3 knows blinded key $BK_{<2,3>}$ (the blind key of newly joined member) and $BK_{<1,0>}$, so M_3 can compute the new group key $K_{<0,0>}$ as it can compute the

intermediate key $K_{<1, 0>}$. Any other member can also compute the new group key after sponsor M_3 publishes the blinded key of $K_{<1, 0>}$. The leave operation is quite similar. See Figure 16 for illustration.
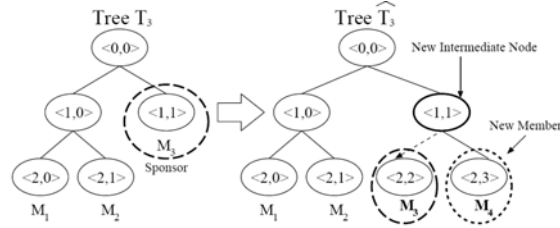

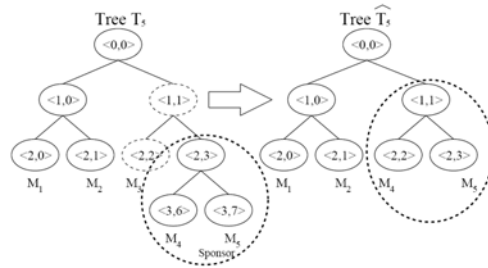Figure 15 Illustration of join member in TGDH

Figure 16 Illustration of leaving member in TGDH

5.4 Group Diffie-Helman (GDH)

GDH is a group key distribution scheme proposed by Steiner, Tsudik, and Waidner [31]. GDH actually contains three key distribution schemes that are extended from the DH protocols. In this article, we only give the algorithm of GDH.3 and ignore the GDH.1 and GDH.2 since these two protocols need a total of $O(n^2)$ exponentiations. The first stage is to collect contributions from all group members (upflow). At the end of this stage user U_{n-1} obtains $g^{\prod\{N_k|k\in[1,n-1]\}}$ and broadcasts this value to all other group members at the second stage. At the third stage, every user U_i ($i \neq n$) factors out its own exponent and forwards the result to the last user U_n. At the final stage, U_n collects all inputs from the previous stage, raises every one of them to the power of N_n and broadcasts the resulting n-1 values to the rest of the group. In the end, every group member has a value of the form $g^{\prod\{N_k|k\in[1,n]\wedge k\neq i\}}$ and can easily compute the group key K_n. Member addition and deletion can be handled easily in this scheme.

A simple example is shown below to illustrate the operation of this scheme for a group of four members, A, B, C, D:

Stage 1: A→ {B}: g^a ; B→{C}: g^{ab}

Stage 2: C →{A, B, D}: g^{abc}

Stage 3: A→{D}: g^{bc} ; B→{D}: g^{ac} ; C→{D}: g^{ab}

Stage 4: D→{A, B, C}: g^{bcd} , g^{acd} , g^{abd} , $\{g^{abc}\}$

Stage 5: $K= g^{abcd}$

The total number of exponentiations of GDH.3 is *5n-6*, the total number of rounds is *n+1*, and the number of messages is *2n-1*.

5.5 Burmester-Desmedt (BD)

BD is a distributed group key management scheme proposed by Burmester and Desmedt [33]. It is an extension of the Diffie-Hellman key distribution system. The core algorithm of this scheme is as follows:

Step 1: Each group member U_i selects a random exponent r_i, and then computes and broadcasts $z_i = g^{r_i} \bmod p$.

Step 2: Each group member U_i computes and broadcasts $X_i = (\frac{z_{i+1}}{z_{i-1}})^{r_i} \bmod p$

Step 3: Each group member U_i computes the common secret,

$k_i = (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} ... X_{i-2} \bmod p$. That is each group user will come up with the same secret $k = g^{r_1 r_2 + r_2 r_3 + ... + r_n r_1} \bmod p$, which is the group key shared by all group members.

In BD scheme, each group member needs to perform *n+1* exponentiations. It also requires a total number of *2n* broadcast messages. Considering a simple example with a group of four users A, B, C, D in the group, user B can compute $k = (g^a)^{4b} \cdot (g^{cb}/g^{ab})^3 \cdot (g^{dc}/g^{bc})^2 \cdot (g^{ad}/g^{cd})^1 = g^{ab+bc+cd+da}$. Obviously, it can be verified that other users A, C, and D can compute the same key as B.

5.6 Skinny Tree (STR)

STR is a simple group key management scheme proposed by Steer and Strawczynski [36]. It is also extended from the DH. STR requires group users to be ordered in a chain. The outline of the algorithm is the following:

Step 1: Every user generates a random number r_i, and broadcasts $g^{r_i} \bmod p$.

Step 2: Users are ordered as a chain. The first and the second user can calculate the value

$$k = g^{(r_n g^{(r_{n-1} g^{(...g^{(r_3 g^{(r_1 r_2)})})})})}$$

However, users *3* to *n* require further information to calculate *k*. The detailed algorithm as given by the author is skipped here, which can be referred in [36]. A simple example of 4 users A, B, C, and D is shown is Figure 17. This scheme takes two rounds and four modular exponentiations, which makes it suited for adding new group members. However, member exclusion is relatively difficult.
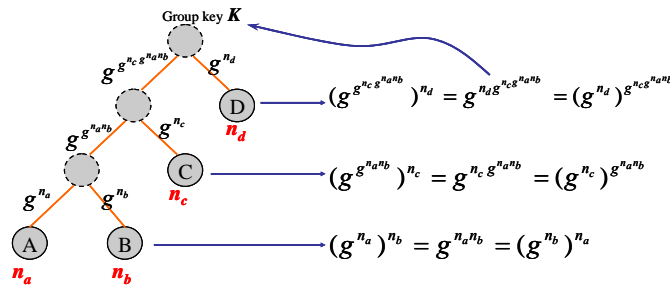

Figure 17: Illustration of STR

6. Open Challenges and Future Directions

Security is an important feature that determines the success and degree of deployment of MANETs. Cryptography is a powerful tool to defend against a variety of attacks and helps to achieve a variety of security goals. Most cryptographic algorithms require the use of keying materials. If the cryptographic key is disclosed, then there is no security at all. Obviously, key management is in the central part of any secure communication and is the weakest point of the security. However, ensuring the security of MANETs is more challenging because of the host mobility, shared wireless medium, resource constraint of physical devices, and most seriously, lack of a fixed and trustable control point in MANETs. Designing and building an underlying secure, robust and scalable key management system is a difficult problem that has received increased attention recently. The current research on key management in MANETs is still at its early stage.

Research on key management in MANETs goes in three directions according to the trust models, which are centralized, decentralized, and fully distributed. While centralized approaches are of least interest in MANETs, decentralized approaches have gained a lot of research attention. The fully distributed trust model is also favored for MANETs. Interestingly, a hybrid approach that combines the centralized model with the distributed scheme has been proposed recently.

Key management in MANETs can also be roughly classified into unicast and multicast key management according to the communication type. Previously, most research focused on the secure pairwise communications, and key management focus was on how to distribute or establish a session key between a pair of communication parties. Currently, secure group communications, such as dynamic conferencing or multicasting in MANETs, is becoming an active research area. The security of group communication involves the management of group keys. For efficiency, tree-based structures are utilized when a central or virtual central control entity is available. Most contributory group key distributions are based on DH protocol with different implementations. Meanwhile, key management can also be classified into symmetric and asymmetric key management depending on the underlying cryptographic algorithms used. Currently, most key management schemes are based on asymmetric cryptosystems. However, for some specific types of MANETs, such as sensor networks, the symmetric key management scheme is dominant. An example of a symmetric approach is the random key pre-distribution in sensor networks.

In summary, based on different assumptions, many key management protocols have been proposed for the MANETs. All key management approaches are subject to various restrictions such as mobile devices available resources, the network bandwidth, and MANETs dynamic nature. An efficient key management protocol for MANETs is an ongoing hot research area.

References

[1] A. Salomaa. Public-Key Cryptography, Springer-Verlag, 1996.
[2] A. Tanenbaum. Computer Networks, PH PTR, 2003.

[3] L. Zhou and Z. Haas. Securing Ad Hoc Networks, IEEE Network Magazine vol.13 No.6 (1999) pp. 24-30.

[4] B. Wu, J. Chen, J. Wu, and M. Cardei. A Survery on Attacks and Countermeasures in Mobile Ad Hoc Networks. Wireless/Mobile Network Security, Springer. Book chapter, Chapter 12, 2006.

[5] S. Yi, P. Naldurg, and R. Kravets. Security Aware Ad Hoc Routing for Wireless Networks. Report No.UIUCDCS-R-2002-2290, UIUC, 2002.

[6] H. Luo and S. Lu. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, IEEE/ACM Transactions on Networking Vol.12 No.6 (2004) pp. 1049-1063.

[7] W. Lou and Y. Fang. A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions. Ad Hoc Wireless Networks, edited by X. Chen, X. Huang and D. Du. Kluwer Academic Publishers, pp. 319-364, 2003.

[8] S. Burnett and S. Paine. RSA Security's Official Guide to Cryptography, RSA Press, 2001.

[9] M. Ilyas. The Handbook of Ad Hoc Wireless Networks, CRC Press, 2003.

[10] S. Yi and R. Kravets. Composite Key Management for Ad Hoc Networks. Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), pp. 52-61, 2004.

[11] W. Mehuron. Digital Signature Standard (DSS). U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL). FIPS PEB 186, 1994.

[12] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang. Security in Mobile Ad Hoc Networks: Challenges and Solutions. IEEE Wireless Communications, pp. 38- 47, 2004.

[13] C. Perkins. Ad Hoc Networks, Addison-Wesley, 2001.

[14] R. Oppliger. Internet and Intranet Security, Artech House, 1998.

[15] B. Wu, J.Wu, E. Fernandez, S. Magliveras, and M. Ilyas. Secure and Efficient Key Management in Mobile Ad Hoc Networks. Proc. of 19th IEEE International Parallel & Distributed Processing Symposium, Denver, 2005.

[16] S. Ravi, A. Raghunathan, and N. Potlapally. Secure Wireless Data: System Architecture Challenges. Proc. of International Conference on System Synthesis, 2002.

[17] B. Wu, J. Wu, E. Fernandez, M. Ilyas, and S. Magliveras. Secure and Efficient Key Management Scheme in Mobile Ad Hoc Networks. Journal of Network and Computer Applications (JCNA), Dec. 2005.

[18] W. Stallings. Wireless Communication and Networks, Pearson Education, 2002.

[19] A. Tanenbaum. Network Security, Chapter 8, Computer Networks. Prentic Hall PTR, 4th Edition, 2002.

[20] C. Murthy and B. Manoj. Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall PTR, 2005.

[21] T. Karygiannis and L. Owens. Wireless Network Security-802.11, Bluetooth and Handheld Devices. National Institute of Standards and Technology. Technology Administration, U.S Department of Commerce, Special Publication. pp. 800-848, 2002.

[22] R. Nichols and P. Lekkas. Wireless Security-Models, Threats, and Solutions, McGraw Hill, Chapter 7, 2002.

[23] C. Kaufman, R. Perlman, and M. Speciner. Network Security Private Communication in a Public World, Prentice Hall PTR, A division of Pearson Education, Inc., 2002

[24] A. Herzberg, S. Jarecki, H. Krawczyk, and H. Yung. Proactive Secret Sharing or: How to Cope With Perpetual Leakage. Proceedings of Crypto'95, vol. 5. 1995. pp. 339–52.

[25] A. Menezes, P. Oorschot, and S. Vanstone. Handbook of Applied Cryptography, CRC Press, 1996.

[26] J. Hubaux, L. Buttyan, and S. Capkun. The Quest for Security in Mobile Ad Hoc Networks, In Proc. of the ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001), Long Beach, CA, October 2001.

[27] A. T. Sherman and D. A. McGrew. Key Establishment in Large Dynamic Groups Using One-Way Function Trees. IEEE Transactions on Software Engineering, vol. 29, no. 5, pp. 444-458, May 2003.

[28] S. Capkun, L. Buttya, and J. P. Hubaux. Self-Organized Public Key Management for Mobile Ad Hoc Networks, IEEE Trans. Mobile Computing, vol. 2, no. 1, pp. 52-64, January-March 2003.

[29] Y. Kim, A. Perrig, and G. Tsudik. Simple and Fault-tolerant Key Agreement for Dynamic Collaborative Groups. Technical Report 2, USC Technical Report 00-737, August 2000.

[30] Y. Kim, A. Perrig, and G. Tsudik. Simple and Fault-tolerant Key Agreement for Dynamic Collaborative Groups. In 7th ACM Conference on Computer and Communications Security, pp. 235-244. ACM Press, November 2000.

[31] M. Steiner, G. Tsudik, and M. Waidner. Cliques: A New Approach to Group Key Agreement. IEEE Transactions on Parallel and Distributed Systems, 2000.

[32] S. Rafaeli and D. Hutchison. A Survey of Key Management for Secure Group Communication. ACM computing Surveys, vol. 35, no. 3, pp. 309-329, September 2003.

[33] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution system. In A. De Santis, editor, Advances in Cryptology – EUROCRYPT '94, no. 950.

[34] C. Wong, M. Gouda, and S. Lam. Secure Group Gommunications Using Key Graphs. In Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, pp. 68–79, 1998.

[35] D. M. Wallner, E. J. Harder, and R. C. Agee. Key Management for Multicast: Issues and Architectures, Internet Draft (work in progress), draft-wallner-key-arch-01.txt, Internet Eng. Task Force, September 1998.

[36] D. Steer, L. Strawczynski, W. Diffie, and M. Wiener. A secure audio teleconference system. In Advances in Cryptology -- CRYPTO'88, August 1990.

[37] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. To appear in Proc. of the IEEE Security and Privacy Symposium, 2003.

[38] W. Du, J. Deng, Y. Han, and P. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In Proc. of 10th ACM Conference on Computer and Communications Security (CCS), Washington DC, October 27-31, 2003.

[39] H. Chan and A. Perrig. PIKE: Peer Intermediaries for Key Establishment in Sensor Networks, In Proceedings of IEEE INFOCOM, March 2005.

[40] A. C-F. Chan, Distributed Symmetric Key Management for Mobile Ad hoc Networks, IEEE INFOCOM, 2004.

[41] A. Shamir. How to Share a Secret. Communications ACM 1979; 22(11), pp. 612–613.

[42] T. Wong, C. Wang, and J. Wing. Verifiable Secret Redistribution for Threshold Sharing Schemes. Technical Report, CMU-CS-02-114-R, School of Computer Science, Carnegie Mellon University, September 2002.

[43] M. Stadler. Publicly Verifiable Secret Sharing. Proceeding of Eurocrypt'96. pp. 190-199, 1996.

[44] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks. Proceeding of The 9th International Conference on Network Protocols, 2001.

[45] B. Wu and J. Wu. A Simple and Efficient Group Key Management Scheme for Mobile Ad Hoc Networks. Submitted to The IEEE International Conference on Communications (ICC 2007). Under review.