IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. XI (Mar-Apr. 2014), PP 146-158
www.iosrjournals.org

Secure and Efficient Key Management Scheme in MANETs

Abu Taha Zamani[1], Syed Zubair[2]
[1]*Lecturer,Deanship of Information Technology,Northern Border University,Kingdom of Saudi Arabia*
[2]*Lecturer,Deanship of Information Technology,Northern Border University,Kingdom of Saudi Arabia*

Abstract: *InMobile ad hoc networks (MANETs) security has become a primary requirements.Thecharacteristics capabilities of MANETsexposeboth challenges and opportunities in achieving key security goals,such as confidentiality,access control,authentication, availability, integrity, and non-repudiation.Cryptographic techniques are widely used for secure communications in both TCP and UDPnetworks. Most cryptographic mechanisms, such as symmetric and asymmetric cryptography,often involve the use of cryptographic keys. However, all cryptographic techniques will beunsecure or inefficient if the key management is weak. Key management is also a central component inMANET security. The main purpose of key management is to provide secure methods for handlingcryptographic keying algorithm. The tasks of key management includes keys for generation, distribution and maintenance. Key maintenance includes the procedures for key storage, keyupdate, key revocation, etc. In MANETs, the computational load and complexityfor key management are strongly subject to restriction by the node's available resources and thedynamic nature of network topology. A number of key management schemes have beenproposed for MANETs. In this article, we present a survey of the research work on keymanagement in MANETs according to recent publications.*
Keywords: *Mobile ad hoc networks, Key management, , Security, PKI, MOCA*

I. Introduction

Key management is a basic part of any secure communication. Most cryptosystems rely on someunderlying secure, robust, and efficient key management system. Secure networkcommunications normally involve a key distribution procedure between communication parties,in which the key may be transmitted through insecure channels. A framework of trustrelationships needs to be built for authentication of key ownership in the key distributionprocedure. While some frameworks are based on a centralized trusted third party (TTP), otherscould be fully distributed. For example, a certification authority (CA) is the TTP in asymmetriccryptosystems, a key distribution center (KDC) is the TTP in the symmetric system, and in PGPno TTP is assumed. According to recent publications, the centralized approach is regarded asinappropriate for MANETs because of the dynamic environment and the transient relationshipsamong mobile nodes. Most researchers prefer the decentralized trust model for MANETs.Several decentralized solutions have been proposed in recent papers with differentimplementations, such as how the CA's responsibility is distributed to all nodes, or to a subset ofnodes.

1.1 Fundamentals of Key Management

Cryptographic algorithms are security primitives that are widely used for the purposes ofauthentication, confidentiality, integrity, and non-repudiation. Most cryptographic systemsrequire an underlying secure, robust, and efficient key management system. Key management isa central part of any secure communication and is the weakest point of system security and theprotocol design.A key is a piece of input information for cryptographic algorithms. If the key was released, theencrypted information would be disclosed. The secrecy of the symmetric key and private keymust always be assured locally. The Key Encryption Key (KEK) approach [8] could be used atlocal hosts to protect the secrecy of keys. To break the cycle (use key to encrypt the data, and usekey to encrypt key) some non-cryptographic approaches need to be used, e.g. smart card, orbiometric identity, such as fingerprint, etc.Key distribution and key agreement over an insecure channel are at high risk and suffer frompotential 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 otherend. In the Diffie-Hellman (DH) scheme [8], the communication parties at both sides exchangesome public information and generate a session key on both ends. Several enhanced DH schemeshave been invented to counter man-in-the-middle attacks. In addition, a multi-way challengeresponse protocol, such as Needham-Schroeder [19], can also be used. Kerberos [19], which isbased on a variant of Needham-Schroeder, is an authentication protocol used in many realsystems, including Microsoft Windows. However, in MANETs, the lack of a central controlfacility, the limited computing resources, dynamic network topology, and the difficulty ofnetwork synchronization all contribute to the complexity of key management protocols.Key integrity and ownership should be protected from advanced key attacks. Digital signatures,hash functions, and the hash function based message authentication code (HMAC) [25] aretechniques used for data authentication and/or integrity purposes. Similarly, the public key isprotected 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 systemslacking 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 isdistributed to a subset or all of the network hosts based on threshold cryptography. Obviously, acertificate cannot prove whether an entity is "good" or "bad". However, it can prove ownershipof a key. Certificates are mainly used for key authentication.A cryptographic key could be compromised or disclosed after a certain period of usage. Since thekey should no longer be usable after its disclosure, some mechanism is required to enforce thisrule. In PKI, this can be done implicitly or explicitly. The certificate contains the lifetime ofvalidity - it is not useful after expiration. However, in some cases, the private key could bedisclosed during the valid period, in which case the CA needs to revoke a certificate explicitlyand notify the network by posting it onto the certificate revocation list (CRL) to prevent itsusage.Key management for large dynamic groups is a difficult problem because of scalability andsecurity. Each time a new member is added or an old member is evicted from the group, thegroup key must be changed to ensure backward and forward security. Backward security meansthat new members cannot determine any past group key and discover the previous groupcommunication messages. Forward security means that evicted members cannot determine anyfuture group key and discover the subsequent group communication information. The group keymanagement should also be able to resist against colluded members.

1.2 Trust Models

1.2.1 Centralized trust model

For the centralized trust model, there is a well-trusted entity known as a TTP [4] [23] [25]. ATTP is an entity trusted by all users in the system, and it is often used to provide keymanagement services. Depending on the nature of their involvement, TTPs can be classified intothree categories: inline, online, or offline. See Figure 1 for an illustration. An inline TTPparticipates actively in between the communication path of two users. An online TTPparticipates actively but only for management purposes, as the two parties communicate with each other directly. An offline TTP communicates with users prior to the setting up ofcommunication links and remains offline during network operation.

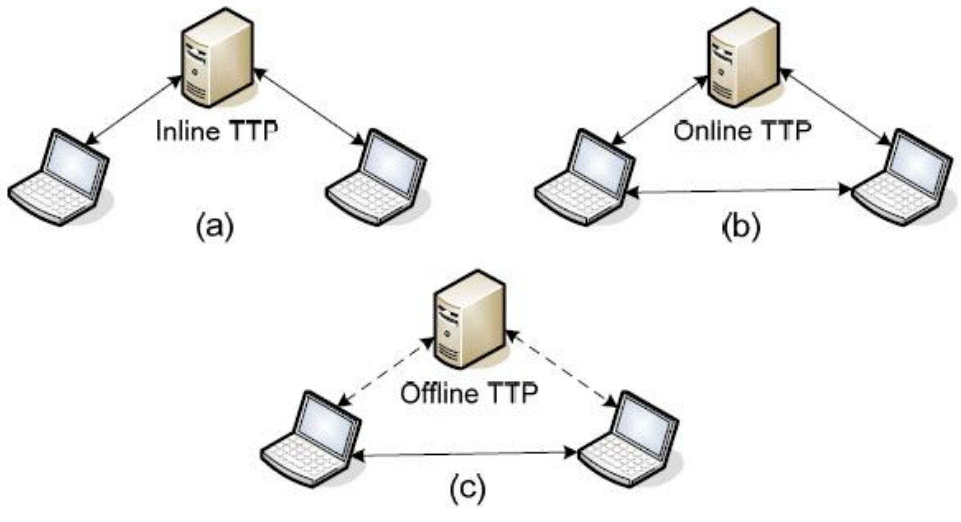

Figure 1: Categories of trust third parties

1.2.1.1 TTPs in symmetric key management systems

TTPs have been implemented in both symmetric and asymmetric key management systems. KeyDistribution Centers (KDC) and Key Translation Centers (KTC) [14] are TTPs in symmetriccryptographic key management systems and the certification authority (CA) is the TTP in publickeymanagement systems. KDC and KTC simplify the symmetric key management since eachuser does not have to share a secret key with every other user. Instead, it only needs to share onekey with the TTP. This reduces the total number of keys that need to be managed from$n(n-1)$ to$n$, where n is the total number of users. Figure 2 illustrates the protocols by implementing KDCor KTC.

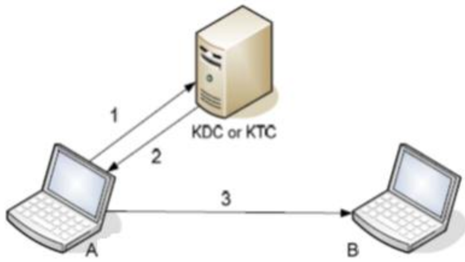

Figure 2: Establishment of session key using KDC or KTC

1 A requests to share a secret key with B. If the TTP is KDC, it generates a key touse. Otherwise, A provides it.The message is encrypted using the secret key sharedbetween A and the TTP.
2 The TTP encrypts the session key with the key it shares with B and returns it toA.
3 A sends the encrypted session key to B, who can decrypt it and thereafter use it tocommunicate securely with A.

1.2.1.2 Public key infrastructure (PKI)

The use of public key cryptography requires the authenticity of public keys. Otherwise, it is easyto forge or spoof someone's public key. Some trusted framework must be present to verify theownership of a public key. A straightforward solution is to have any two users that wish tocommunicate exchange their public keys in an authenticated manner. It would require the initialdistribution of *n(n-1)* public keys. Obviously, this solution is not scalable for a large network andhas the same problems we discussed in the symmetric key management system. However, byhaving a trusted third party issue certificates to each of the users, every user only needs to holdthe 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 andweb-of-trust trust models [4] [10]. For network scalability, the centralized trust model could be ahierarchical trust structure instead of a single CA entity. Multiple CA roots could be necessaryfor a large network, such as the Internet. We will discuss the fully distributed or web-of-trustmodel later.A PKI provides the mechanisms needed to manage certificates, and normally consists of thecomponents illustrated in Figure 3.

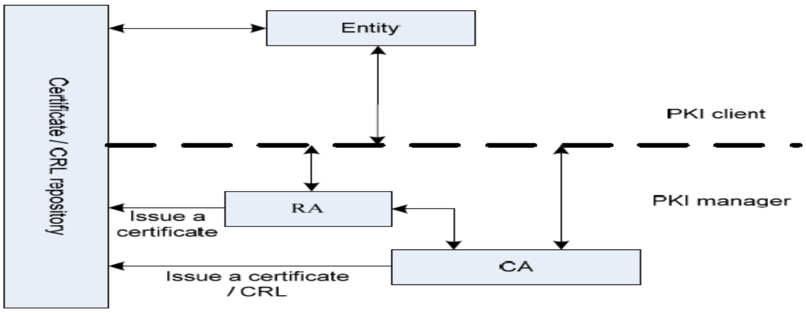

Figure 3: Components of a PKI

In this diagram, the certification authority (CA) is the component responsible for issuing andrevoking certificates, while the registration authority (RA) is responsible for establishing theidentity 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, keyupdate, revocation, key recovery, cross-certification, etc.

1.2.2 Web-of-trust model

The web-of-trust model is also called certificate chaining. PGP [19] is an example built on thistrust 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 centralizedtrust model, the web-of-trust model does not require a heavy infrastructure or complexbootstrapping procedures, and every node plays an identical role and shares the sameresponsibility. Although the web-of-trust model has the above advantages, it has two majorlimitations. First, a certificate graph may not populate enough to provide certificate chains for agiven 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 thecorrect behaviors of all participants. Obviously, that cannot always be assumed. However, sincethere is no clear way to tell if a certificate chain includes any misbehaving nodes, the overallconfidence for the certificate is relatively low.

1.2.3 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 becausethe central entity could become a hot spot of attacks. Thus, this network suffers from a securitybottleneck. Meanwhile a completely distributed model may not be acceptable because there is nowell-trusted security anchor available in the whole system. One feasible solution is to distributethe central trust to multiple entities (or the entire network) based on a secret sharing scheme. Inthe decentralized public key management scheme, the system public key is distributed to theentire 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 incombination.

1.2.4 Hybrid trust model

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

II. Overview of Key Management Schemes in MANETs

2.1 Asymmetric key management schemes

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

2.2 Symmetric key management schemes

There are research papers that are based on the symmetric-key cryptography for securingMANETs. For instance, some symmetric key management schemes are proposed for sensornodes that are assumed to be incapable of performing costly asymmetric cryptographiccomputations. Pairwise keys can be preloaded into nodes, or based on the random keydistribution in which a set of keys is preloaded. Chan introduced a distributed symmetrickey distribution scheme for MANETs. The basic idea is that each node is preloaded with a set ofkeys from a large key pool. The key pattern should satisfy the property that any subsetof nodes can find at least one common key, and the common key should not be covered by acollusion of a certain number of other nodes outside the subset. Chan and Perrig introduceda symmetric key agreement scheme for the sensor nodes. The basic idea of their approach is thateach node shares a unique key with a set of nodes vertically and horizontally (in 2-Dimensions).Therefore, any pair of nodes can rely on at least one intermediate node to establish the commonkey.

2.3 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 scalabilityand security. For instance, each time a new member is added or an old member is evictedfrom a group, the group key must be changed to ensure backward and forward security.

III. Asymmetric Key Management Schemes in MANETs

3.1 Secure Routing Protocol (SRP)

SRP is a decentralized public key management protocol proposed by Zhou and Hass [3] byemploying *(t, n)* threshold cryptography in their research paper called "SecuringAd Hoc Networks". In the system, there are *n* servers, which are responsible for public-keycertificate services. Therefore, the system can tolerate *t-1* compromised servers. Servers canproactively refresh the secret shares using the proactive secret sharing (PSS) [24] techniques orby adjusting the configuration structure based on share redistribution techniques to handlecompromised 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 shortamount of time, which obviously increases the difficulty of the success of adversaries. Thesystem configuration of this scheme is illustrated in Figure 4. The system public key K isdistributed to all nodes in the network, whereas the private key S is split to *n* shares s1, s2, s3, …,sn, one share for each server according to a random polynomial function.

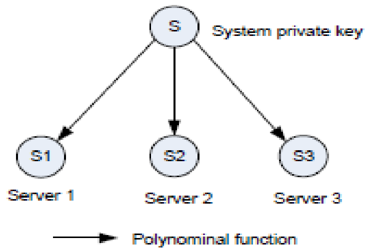

Figure 4: Illustration of SRP scheme

In this scheme, the system model is such that *n* servers are special nodes, each with its ownpublic/private key pair and the public key of every node in the network. This is a critical issue ina large network. However, this scheme does not describe how a node can contact *t* serverssecurely and efficiently in case the servers are scattered in a large area. A share-refreshingscheme is proposed to counter mobile adversaries. The update of secret shares does not changethe system public/private key pairs. Therefore, nodes in the network can still use the samesystem public key to verify a signed certificate so that the share-refreshing is transparent to allnodes. However, a method of distributing these updated sub shares to all nodes securely andefficiently 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] intheir paper "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks". TheURSA protocol is also based on threshold cryptography as in SRP [3]. The difference betweenURSA and SRP is that in URSA, all nodes are servers and are capable of producing a partialcertificate, while in SRP only server nodes can produce certificates. Thus, certificate services aredistributed to all nodes in the network. URSA also proposed a distributed self-initialization phasethat allows a newly joined node to obtain secret shares by contacting a coalition of *k* neighboringnodes without requiring the existence of an online secret share dealer. The basic idea is to extendthe PSS technique by shuffling the partial shares instead of shuffling the secret sharingpolynomials. The purpose of this shuffling process is to prevent deducing the original secretshare from a resulting share.In URSA, every node should periodically update its certificate. To update its certificate, a nodemust 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 willintroduce either communication delays or cause search failures. It could potentially utilizeservices from 2-hop neighboring nodes.

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

3.3 Mobile Certificate Authority (MOCA)

MOCA is a decentralized key management scheme proposed by Yi, Naldurg, and Kravets [5] intheir paper "Key management for heterogeneous ad hoc wireless networks". In this approach, acertificate service isdistributed to Mobile Certificate Authority (MOCA) nodes. MOCA nodesare chosen based on heterogeneity if the nodes are physically more secure and computationallymore powerful. In cases where nodes are equally equipped, they are selected randomly from thenetwork. The trust model of this scheme is a decentralized model since the functionality of CA isdistributed to a subset of nodes. A service-requesting node can locate $k + \alpha$ MOCA nodes eitherrandomly, 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 securerouting 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 intheir paper “Self-organized public key management for mobile ad hoc networks”. This scheme isbased on the web-of-trust model that is similar to PGP [8]. The basic idea is that each user acts asits own authority and issues public key certificates to other users. A user needs to maintain twolocal certificate repositories. One is called the non-updated certificate repository and the otherone is called the updated certificate repository. The reason a node maintains a non-updatedcertificate repository is to provide a better estimate of the certificate graph. Key authentication isperformed via chains of public key certificates that are obtained from other nodes throughcertificate exchanging, and are stored in local repositories.

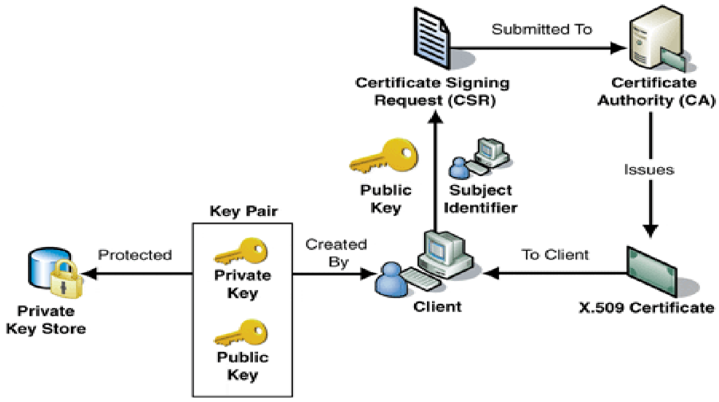

Figure 5: An example of certificate chain

The fully distributed, self-organized certificate chaining has the advantage of configurationflexibility and it does not require any bootstrapping of the system. However, this certificatechaining requires a certain period to populate the certificate graph. This procedure completelydepends on the individual node’s behavior and mobility. One the other hand, this fully self-organizedscheme lacks any trusted security anchor in the trust structure that may limit its usagefor applications where high security assurance is demanded. In addition, many certificates needto 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 stronglyconnected, especially in the mobile ad hoc scenario. In that case, nodes within one componentmay not be able to communicate with nodes in different components. Certificate conflicting isanother potential problem in this scheme.

3.5 Composite Key Management

Recently, Yi, and Kravets [10] provided a composite key management scheme in their paper“Composite key management for ad hoc networks”. In their scheme, they combine thecentralized trust and the fully distributed certificate chaining trust models. This scheme takesadvantage 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 thatcomprise the virtual CA. Some authentication metrics, such as confidence value, are introducedin order to “glue” two trusted systems. A node certified by a CA is trusted with a higherconfidence level. However, properly assigning confidence values is a challenging task. Anexample of a composite key management model is shown in Figure 6.

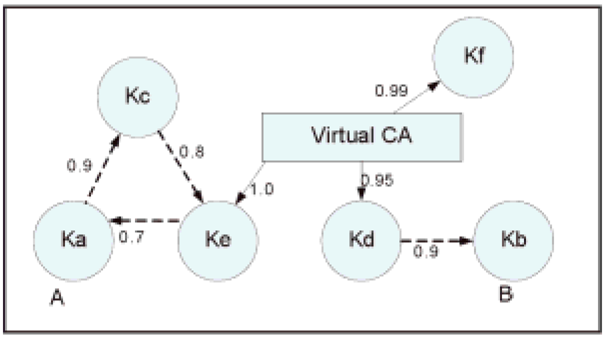

Figure 6: An example of composite key management scheme

3.6 Secure and Efficient Key Management (SEKM)

SEKM is a decentralized key management scheme proposed by Wu and Wu [15] [17] in theirpaper "Secure and efficient key management in mobile ad hoc networks". It is based on thedecentralized virtual CA trust model. All decentralized key management schemes are quitesimilar in that the functionality of the CA is distributed to a set of nodes based on the techniquesof threshold cryptography. However, no schemes except for SEKM present detailed, efficient,and secure procedures for communications and cooperation between secret shareholders thathave more responsibilities. In SEKM, all servers that have a partial system private key are toconnect and form a server group. The structure of the server group is a mesh structure as shownin Figure 7. Periodic beacons are used to maintain the connection of the group so servers canefficiently coordinate with each other for share updates and certificate service. The problem withSEKM is that, for a large network with highly dynamic mobility, maintaining the structure servergroup can be costly.

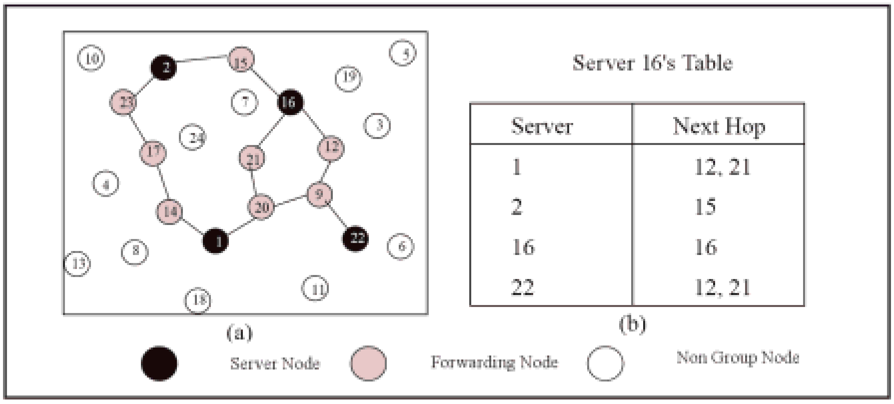

Figure 7: Server group structure in SEKM

IV. 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 in the paper"Distributed symmetric key management for mobile ad hoc networks". It is aimed at the networksettings where mobile nodes are not assumed to be capable of performing computationallyintensive public key algorithms and the TTP is not available. The basic idea of the DKPS schemeis that each node randomly selects a set of keys in a way that satisfies the probability property ofcover-free family (CFF). Any pair of nodes can invoke the secure shared key discoveryprocedure (SSD). The theory behind the SSD is the additive and scalar multiplicativehomomorphism 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 sendthe encrypted polynomial to the other side. The coefficients of the polynomial are encrypted withthe sender's secret key. The other side will send back the encrypted polynomial multiplied by arandom value. Because of the homomorphism and non-trivial zero encryption properties, eitherside 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 in theirpaper "PIKE: Peer intermediaries for key establishment in sensor networks". It is a random keypre-distribution scheme. The basic idea of PIKE is to use sensor nodes as trusted intermediariesto establish shared keys. Each node shares a unique secret key with a set of nodes. In the case of2-Dimension, a node shares a unique secret with each of theO(*n*) nodes in the horizontal andvertical dimensions. Therefore, any pair of nodes can have a common secret with at least oneintermediate node. This key pre-distribution scheme can be extended to three or moredimensions. Figure 8 shows the basic idea of the PIKE scheme. Dark lines connect the nodes thatshare a unique key with node A, and light lines connect nodes that share a unique key with nodeB. There are six nodes that each share a unique key with node A and node B.

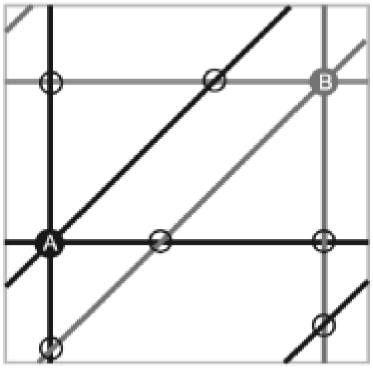

Figure 8: Illustration of PIKE scheme

V. Group Key Management Approaches

The messages are protected by encryption using the chosen key, which in the context of groupcommunication is called the group key. Only those who know the current group key are able torecover the original message. Group key establishment means that multiple parties want to createa common secret to be used in the secure exchange of information. Two people who did notpreviously share a common secret can create one common secret with a DH key exchangeprotocol. The 2-party DH protocol can be extended to a generalized version of the *n*-party DHkey-exchange model. Research efforts have been put into the design of group key agreementprotocols to achieve better scalability, efficiency, and storage saving, such as the introduction ofa tree structure and hash function. Furthermore, the group key management also needs to addressthe security issue related to membership changes. The modification of membership could requirethe group key to be refreshed (e.g., periodic re-key). The change of group keys when oldmembers leave or new members join ensures backward and forward security. Therefore, a groupkey 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. In centralized group key protocols, a single entityis employed to control the whole group and is responsible for re-keying and distributing groupkeys to group members. In the decentralized approaches, a set of group managers is responsiblefor managing the group as opposed to a single entity being held responsible. In the distributedmethod, group members themselves contribute to the formation of group keys and are equallyresponsible for the re-keying and distribution of group keys. Recently, collaborative and grouporientedapplications in MANETs have become an active research area. Obviously, group keymanagement is a central building block in securing group communications in MANETs.However, group key management for large and dynamic groups in MANETs is a difficultproblem 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 anoverview of those protocols. Most of the following group key protocols are designed for theinfrastructure networks. However, with the proper extension, some of them could be utilized andadapted to the MANET environment, or could serve as a hint for the design of MANET-specificgroup key management protocols. For instance, GDH (Section 5.4) and LKH (Section 5.1) havebeen extended into the MANETs. proposed a simple and efficient group key managementscheme, called SEGK, for MANETs. The basic idea of SEGK is that a physical multicast tree isformed in MANETs for efficiency. Group members take turns acting as group coordinator tocompute and distribute intermediate key materials to group members. The keying materials aredelivered through the tree links. The coordinator is also responsible for maintaining theconnection of the multicast group. All group members can compute the group key locally in adistributed manner.

5.1 Logical Key Hierarchy (LKH)

LKH is a centralized group key management scheme proposed by Wallner, Harder and Agee. It is based on the tree structure with each user (group participant) corresponding to a leafand the group initiator as the root node. The tree structure will significantly reduce the numberof broadcast messages and storage space for both the group controller and group members. Theoperation of this scheme is outlined below.Each leaf node shares a pairwise key with the root node as well as a set of intermediate keysfrom 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. SeeFigure 9: U_5 stores *k5, k56, k58,* and *k0*.

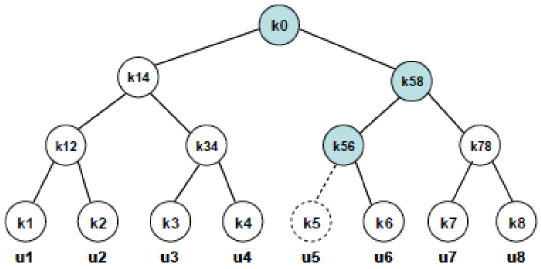

Figure 9: A sample tree structure of LKH

When a member joins the group, the re-key procedure will be started. A re-key message isgenerated containing the new set of keys encrypted with its respective node's children key.Figure 9 shows keys that are affected. The new member U5 receives a secret key *k*5 andattaches the intermediate node k56 logically. The keys *k*56, *k58* and *k0*, which are in the pathfrom *k5* to *k0,* need to be refreshed. New keys, *k*'56, *k'58*, and *k'*0, are generated as illustrated inFigure 10 (a). These keys are encrypted with their respective node's children's key, e.g., oneinstance of *k'56* is encrypted by *k5*, and the other copy is encrypted by *k6*. Theremoval of a member follows a similar procedure. For instance, when member U6 leaves thegroup, *k56*, *k58*, and *k0* should be changed and the new set of keys *k'56*, *k'58*, and *k'0* areencrypted with their respective children's key. See Figure 10 (b) for an illustration of a memberleave.

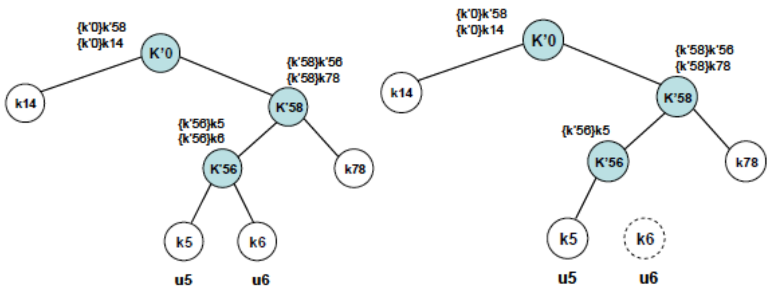

Figure 10 (a) : Illustration of joining member U5

Figure 10 (b) : 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. It is based on the tree structure that is similar to the above LKH scheme. However, all keysin the OFT scheme are functionally related according to a one-way hash function. The idea isthat the keys held by a node's children are blinded using a one-way hash function and thencombined together using a mixing function, such as a bitwise exclusive-or operation. Each groupuser receives blind keys from its sibling set as well as the blind key of its own sibling. Based oncollected blinded keys, the group users can deduce each key of its ancestor set. See Figure 11 foran 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 $\log_2$n is the height of the tree, and *n* is thetotal number of group members. The scheme has the same complexity as the LKH scheme for abalanced tree structure, but in the re-keying process, the size of keying materials reduces from2.$\log_2$n to $\log_2$n.

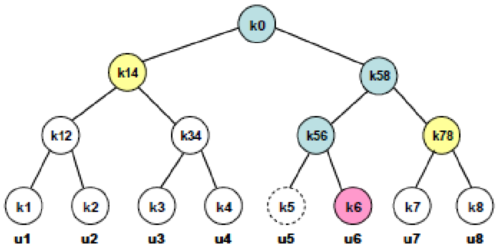

Figure 11: A sample tree structure of OFT

The message size reduction is achieved because in the OFT scheme, the blinded key changed ina node is encrypted only with the key of its sibling node while in LKH scheme the new key mustbe encrypted with its two children’s keys, see Figure 12.

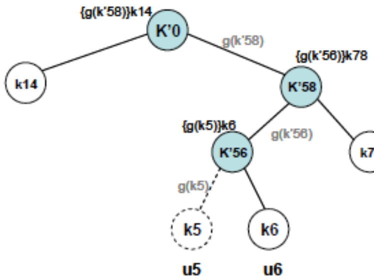

Figure 12: Illustration of join member U5 in OFT

5.3 Tree-Based Group Diffie-Hellman (TGDH)

TGDH is a group key management scheme proposed by Kim, Perrig, and Tsudik. It is atree-based group DH scheme. The basic idea is to combine the efficiency of the tree structurewith the contributory feature of DH.The basic operation of this scheme is as follows. Each group member contributes its (equal)share to the group key, which is computed as a function of all the shares of current groupmembers. As the group grows, new members' shares are factored into the group key but oldmembers' shares remain unchanged. As the group shrinks, departing members' shares areremoved from the new key and at least one remaining member changes its share. All protocolmessages are signed by the sender using RSA.In TGDH, a sponsor takes a special role that can involve computing keys and broadcasting theblinded keys to the group during events of member join, leave, partition, and merge. Anymember in the group can take on this responsibility. Figure 13 (a) illustrates the operation ofmember join. When M4 joins the group, sponsor M3 will rename node <1, 1> to <2, 2>; generatea 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 M3 knows blinded key BK<2, 3> (the blind key of newlyjoined member) and BK<1, 0>, so M3 can compute the new group key K<0, 0> as it can compute theintermediate key K<1, 0>. Any other member can also compute the new group key after sponsorM3 publishes the blinded key of K<1, 0>. The leave operation is quite similar. See Figure 13 (b) for anillustration.

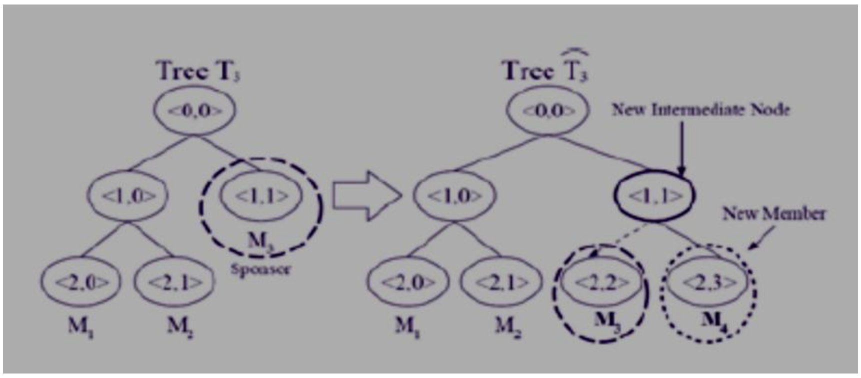

Figure 13 (a) : Illustration of join member in TGDH

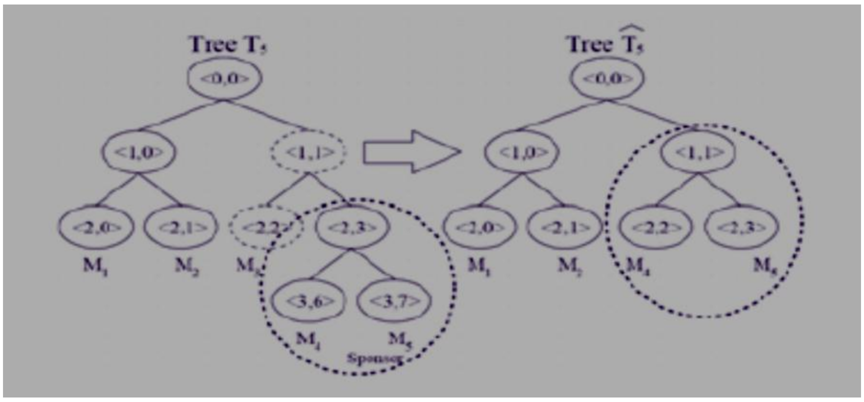

Figure 13 (b) : Illustration of leaving member in TGDH

5.4 Group Diffie-Helman (GDH)

GDH is a group key distribution scheme proposed by Steiner, Tsudik, and Waidner [31]. GDHactually contains three key distribution schemes that are extended from the DH protocols. In thisarticle, we only give the algorithm of GDH.3 and ignore GDH.1 and GDH.2 since these twoprotocols need a total of *O(n2)* exponentiations. The first stage involves collecting contributionsfrom all group members (upflow). At the end of this stage, user U_{n-1}obtains$g^{\Pi\{Nk|k\in[1,n-1]\}}$andbroadcasts this value to all other group members at the second stage. At the third stage, everyuser $U_i(i \neq n)$ factors out its own exponent and forwards the result to the last user U_n. At thefinal stage, $U_n$collects all inputs from the previous stage, raises every one of them to the powerof $N_n$and broadcasts the resulting n-1 values to the rest of the group. In the end, every groupmember has a value of the form $g^{\Pi\{Nk|k\in[1,n]\wedge k \neq i\}}$ 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 fourmembers, 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*, andthe number of messages is *2n-1*.

5.5 Burmester-Desmedt (BD)

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

Step 1: Each group member $U_i$selects a random exponent r_i, and then computes andbroadcasts $z_i = g^{ri} \bmod p$
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 *Ui*computes the common secret, $k_i = (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \dots X_{i-2} \bmod p$. That is each group user will come upwith the same secret $k=g^{r_1r_2+r_2r_3+\dots+r_nr_1} \bmod p$, which is the group key shared byall group members.

In BD scheme, each group member needs to perform *n+1* exponentiations. It also requires a totalnumber 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. It isalso extended from the DH. STR requires group users to be ordered in a chain. The outline ofthe algorithm is the following:

Step 1: Every user generates a random number r_i and broadcasts g^{ri}mod *p* .
Step 2: Users are ordered as a chain. The first and the second user can calculate the value $k = g^{(r_n g^{(r_{n-1} g^{(\dots g^{(r_2 g^{(r_1 r_2)})})})})}$

However, users *3* to *n*require further information to calculate *k*. A simple example of 4 users A, B, C,and D is shown is Figure 14. This scheme takes two rounds and four modular exponentiations,which makes it suited for adding new group members. However, member exclusion is relativelydifficult.

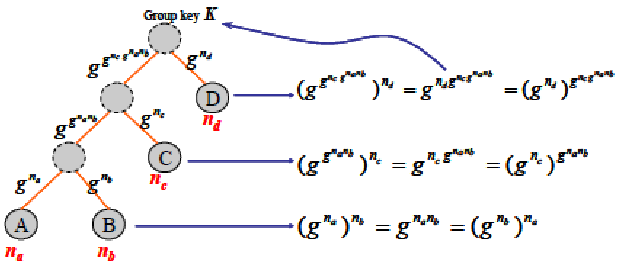

Figure 14: Illustration of STR

VI. Conclusion

Security is an important feature that determines the success and degree of deployment ofMANETs. Cryptography is a powerful tool to defend against a variety of attacks and helps toachieve a variety of security goals. Most cryptographic algorithms require the use of keyingmaterials. If the cryptographic key is disclosed, then there is no security at all. Obviously, keymanagement is in the central part of any secure communication and is the weakest point of thesecurity. However, ensuring the security of MANETs is more challenging because of the hostmobility, 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 underlyingsecure, robust, and scalable key management system is a difficult problem that has receivedincreased attention recently. The current research on key management in MANETs is still at itsearly 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 ofleast interest in MANETs, decentralized approaches have gained a lot of research attention. Thefully distributed trust model is also favored for MANETs. Interestingly, a hybrid approach thatcombines the centralized model with the distributed scheme has been proposed recently.Key management in MANETs can also be roughly classified into unicast and multicast keymanagement according to the communication type. Previously, most research focused on thesecure pairwise communications, and key management focus was on how to distribute orestablish a session key between a pair of communication parties. Currently, secure groupcommunications, such as dynamic conferencing or multicasting in MANETs, is becoming anactive research area. The security of group communication involves the management of groupkeys. For efficiency, tree-based structures are utilized when a central or virtual central controlentity is available. Most contributory group key distributions are based on DH protocol withdifferent implementations. Meanwhile, key management can also be classified into symmetricand asymmetric key management depending on the underlying cryptographic algorithms used.Currently, most key management schemes are based on asymmetric cryptosystems. However, forsome specific types of MANETs, such as sensor networks, the symmetric key managementscheme is dominant. An example of a symmetric approach is the random key pre-distribution insensor networks.In summary, based on different assumptions, many key management protocols have beenproposed for MANETs. All key management approaches are subject to various restrictions suchas the mobile device's available resources, the network bandwidth, and MANETs dynamicnature. An efficient key management protocol for MANETs is an ongoing hot research area.

References

[1] Saloma, A. (1996). Public-Key Cryptography, Springer-Verlag.
[2] Tanenbaum, A. (2003). Computer Networks, PH PTR.
[3] Zhou, L. and Haas, Z. (1999). Securing Ad Hoc Networks, IEEE Network Magazine vol.13, no. 6, pp.24-30.
[4] Wu, B., Chen, J., Wu, J., and Cardei, M. (2006). A Survey on Attacks and Countermeasures in MobileAd Hoc Networks. Wireless/Mobile Network Security, Springer. Chapter 12.
[5] Yi, S., Naldurg, P., and Kravets, R. (2002). Security Aware Ad Hoc Routing for Wireless Networks.Report No.UIUCDCS-R-2002-2290, UIUC.
[6] Luo, H. and Lu, S. (2004). URSA: Ubiquitous and Robust Access Control for Mobile Ad HocNetworks, IEEE/ACM Transactions on Networking. vol. 12, no. 6, pp. 1049-1063.
[7] Lou, W. and Fang, Y. (2003). A Survey of Wireless Security in Mobile Ad Hoc Networks: Challengesand Available Solutions. Ad Hoc Wireless Networks, edited by X. Chen, X. Huang and D. Du.Kluwer Academic Publishers, pp. 319-364.
[8] Burnett, S. and Paine, S. (2001). RSA Security's Official Guide to Cryptography, RSA Press.
[9] M. Ilyas. The Handbook of Ad Hoc Wireless Networks, CRC Press, 2003.
[10] Yi, S. and Kravets, R. (2004). Composite Key Management for Ad Hoc Networks. Proc. of the 1stAnnual International Conference on Mobile and Ubiquitous Systems: Networking and Services(MobiQuitous'04), pp. 52-61.

[11] Mehuron, W. (1994). Digital Signature Standard (DSS). U.S. Department of Commerce, NationalInstitute of Standards and Technology (NIST), Information Technology Laboratory (ITL). FIPS PEB186.

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

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

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

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

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

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

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

[19] Tanenbaum, A. (2002). Network Security, Chapter 8, Computer Networks. Prentice Hall PTR, 4thEdition.

[20] Murthy, C. and Manoj, B. (2005). Ad Hoc Wireless Networks: Architectures and Protocols, PrenticeHall PTR.

[21] Karygiannis, T. and Owens, L. (2002). Wireless Network Security-802.11, Bluetooth and HandheldDevices. National Institute of Standards and Technology. Technology Administration, U.SDepartment of Commerce, Special Publication. pp. 800-848.

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

[23] Kaufman, C., Perlman, R., and Speciner, M. (2002). Network Security Private Communication in aPublic World, Prentice Hall PTR.

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

[25] Bing Wu, Jie Wu and MihaelaCardei, "A Survey of Key Management in Mobile Ad Hoc Networks" , http://www.researchgate.net/publication/228910095