A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks

JOHANN VAN DER MERWE, DAWOUD DAWOUD, and STEPHEN McDONALD

University of KwaZulu-Natal

The article reviews the most popular peer-to-peer key management protocols for mobile ad hoc networks (MANETs). The protocols are subdivided into groups based on their design strategy or main characteristic. The article discusses and provides comments on the strategy of each group separately. The discussions give insight into open research problems in the area of pairwise key management.

Categories and Subject Descriptors: C.2.2 [**Processor Architectures**]: Network Protocols; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms: Design, Management, Security

Additional Key Words and Phrases: Mobile ad hoc networks, security, peer-to-peer key management, pairwise key management

ACM Reference Format:

van der Merwe, J., Dawoud, D., and McDonald, S. 2007. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv. 39*, 1, Article 1 (April 2007), 45 pages DOI = 10.1145/1216370.1216371 http://doi.acm.org/10.1145/1216370.1216371

1. INTRODUCTION

As a result of significant advances in mobile computing and wireless communication technology, mobile devices have gained sufficient communication, computation, and memory resources to be interconnected. By definition, mobile ad hoc networks (MANETs) differentiate themselves from existing networks by the fact that they rely on no fixed infrastructure [Zhou and Haas 1999]: the network has no base stations, access points, remote servers, etc. All network functions are performed by the nodes forming the network; each node performs the functionality of host and router, relaying data to establish connectivity between source and destination nodes not directly within each other's transmission range. This feature makes ad hoc networks financially viable since there is no cost involved in setting up or maintaining a fixed network architecture.

MANETs are autonomous, multihop networks interconnected via wireless links. The word *ad hoc* (translated as *for this only* from Latin) implies that the network is formed in a spontaneous manner to meet an immediate demand and specific goal. MANETs

This research was sponsored by ARMSCOR, the Armaments Corporation of South Africa.

Authors' addresses: School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal, King George V Avenue, Durban, South Africa, 4041; email: {vdmerwe,dawoudd,mcdonalds}@ukzn.ac.za.

have a *dynamic* network topology. Since nodes in the networks are mobile, with unrestricted movement, the configuration of the network topology can change very rapidly. The position of the nodes relative to each other may exhibit a randomly changing nature and therefore be considered as unpredictable. The lack of infrastructure, dynamic network topology, and errorprone wireless connectivity result in frequent link breakages, implying sporadic connectivity. Protocols for MANETs therefore need to mitigate the unreliability of basic network services by taking on a fully distributed, self-organizing nature. From a security perspective, distributing the functionality of network services to as many nodes as possible avoids a single point of attack.

Considering the unique features of ad hoc networks, it is expected that the mechanisms proposed to guarantee the security of conventional wireline networks are not necessarily suitable or adaptable to MANETs. Special mechanisms and protocols designed specifically for ad hoc networks are necessary.

The primary differentiation between existing peer-to-peer key management schemes can be made based on the assumption of an offline and/or online trusted authority.

—*Fully self-organized* MANETs do not have any form of online or offline authority [Capkun et al. 2003b, 2006]. These MANETs are created solely by the end-users in an ad hoc fashion. Fully self-organized MANETs can be informally visualized as a group of strangers, people who have never met before, coming together for a common purpose. The users forming the MANET have no preestablished relationships and therefore share no common keying material on their nodes. Users therefore have to set up security associations between themselves, after network formation, without the aid of a common offline trusted third party (TTP). Furthermore, once the network is operational, there is no form of online TTP to perform any key management services. The network is therefore operated and managed by the nodes themselves, which makes MANETs dependent upon the cooperative and trusting nature of nodes [Buttyan and Hubaux 2003]. The autonomous framework of mobile ad hoc networks or alternatively stated, the requirement that the network is operated primarily by the end-users, is generally referred to as *self-organization* [Capkun et al. 2003b, 2006].

—*Authority-based* MANETs support applications that demand the use of an offline authority [Capkun et al. 2006]. In contrast to fully self-organized MANETs, the nodes in authority-based MANETs do have preestablished relationships. The trusted authority sets up the nodes prior to network formation, that is, provides each node with (shared) cryptographic keying material and a set of (universal) system parameters. During the subsequent online operations, the keying material can be used to establish strong security associations between the nodes [Capkun et al. 2006].

Although authority-based MANETs are not formed purely ad hoc, they share the main characteristics of MANETs such as node mobility and lack of infrastructure. This article expands the notion of *authority-based* MANETs [Capkun et al. 2006] to also include MANETs that use a distributed *online* authority (in addition to the offline authority). The distributed online authority may provide essential certification services such as certificate renewal and revocation.

This survey aims to provide an overall perspective on the area of peer-to-peer key management for MANETs; the scope of the discussion includes key management schemes for fully self-organized MANETs and authority-based MANETs.

The survey is organized as follows: in Section 2 an overview of the characteristics of mobile ad hoc networks is given. Section 3 provides examples of possible applications for ad hoc networks. In Section 4 the need for key management in MANETs is highlighted and the requirements for key management schemes are stipulated. In Section 5 peer-to-peer key management is introduced and the reader is guided into the key management protocol analysis presented in Sections 6 through 13. The article is concludes in Section 14.

2. CHARACTERISTICS OF MOBILE AD HOC NETWORKS

It is important to acknowledge the properties or characteristics of mobile ad hoc networks (MANETs), since these properties can also be seen as *constraints* faced by researchers when designing security protocols for MANETs. Although these constraints are detailed in various articles [Chan 2004; Capkun et al. 2003b; Haas and Tabrizi 1998; Haas et al. 2002], security protocols are frequently published that do not adhere to the fundamental design constraints. These constraints form the basis of protocol analysis. When a novel protocol or scheme is published for MANETs, the feasibility of the proposal is measured by (1) the degree to which the protocol satisfies the fundamental constraints of MANETs and (2) whether the protocol makes a justifiable tradeoff between security, memory requirements, and computational/communication overhead.

Note that not all MANETs adhere to all of the characteristics detailed in this section. The characteristics of MANETs and the possible applications are strongly related; different applications demand MANETs with variants of the given characteristics. For example, an "open" or public MANET will take on a *self-organized* nature, and hence the end-users will set up and manage the network themselves. This means that an offline authority may not be available. In contrast, MANETs used in military applications will not have a self-organized characteristic, but will make use of an offline authority to initialize the nodes; the *authority-based* approach allows for robust access control to the network services.

Another example of varying characteristics emerges from MANETs formed by sensor nodes or laptop computers. Clearly schemes designed for MANETs formed by laptop computers will not have the same limitation on memory, energy (battery), and computational resources as those formed by sensor nodes.

It is thus apparent that a clear description of a key management scheme's intended application is necessary. The application may dictate the characteristics of the MANET and the degree to which some characteristics will influence the design of a suitable scheme.

2.1. Network Infrastructure

There is no fixed or preexisting infrastructure in an ad hoc network: all network functions (routing, security, network management, etc.) are performed by the nodes themselves. Due to the nodes' limited transmission range, data dissemination is achieved in a multihop fashion; nodes can therefore be considered as hosts and routers. Although the lack of infrastructure opens a new window of opportunity for attacks, the authors believe that the lack of infrastructure can help to ensure the survivability of the network in a very hostile environment. This holds true not only from a network security perspective, but also when the users of the network are under physical attack (see Section 3.1).

Ad hoc networks may be spontaneously formed with no a priori knowledge of the physical location and networking environment. MANETs' lack of infrastructure thus makes it suitable for various applications where conventional networks fall short (see Section 3).

Some researchers have already addressed security issues in *hybrid* ad hoc networks (for example, Salem et al. [2005]). Hybrid ad hoc networks combine conventional network infrastructure with multihopping. This derivative of ad hoc networks will find useful application where fixed infrastructure can be extended through multihop networks or where the functionality (and performance) of multihop networks can be enhanced by relying on some infrastructure.

2.2. Network Topology

Nodes in ad hoc networks may be mobile resulting in a dynamic, weakly connected topology. Since node mobility is unrestricted, the topology may be unpredictable. The network will, however, demonstrate global mobility patterns which may not be completely random [Capkun et al. 2006].

The topology is weakly connected due to transient, errorprone, wireless connectivity. The users may therefore experience unavailability of essential networking services. Node mobility and wireless connectivity allow nodes to spontaneously join and leave the network, which makes the network amorphous. Security services must be able to scale seamlessly and remain available with changes in network topology.

2.3. Self-Organization

MANETs cannot rely on any form of central administration or control; this is essential to avoid a single point of attack [Zhou and Haas 1999]. A *self-organized* MANET cannot rely on any form of *offline* trusted third party (TTP); the network can thus be initialized by a distributed *online* TTP.

A *pure* or *fully self-organized* MANET does not rely on any form of TTP whatsoever [Capkun et al. 2003b, 2006], that is, the *online* TTP is also eliminated. Nodes will therefore only have compatible devices with the same software installed. In the extreme case, the nodes will not even share a common set of security system parameters. The lack of a TTP may force the end-users to actively participate in the set up of security associations. A (fully) self-organized MANET has some inherent security implications:

—*Fully* self-organized MANETs are "open" in nature: similar to the Internet, any user can join the network at random. Access control to *applications* will have to be provided at the application layer with a varying degree of user interaction.

—Each user will be its own authority domain, and hence responsible for generating and distributing its own keying material. As pointed out by Douceur [2002], any node can generate more than one identity when there is no offline TTP. It is thus clear that it will be very difficult (if not impossible) to limit users to one and only one unique identity in a (fully) self-organized setting.

—The network will always be vulnerable to the active insider adversary. In fact, the Dolev-Yao adversary model [Dolev and Yao 1983] is too restrictive [Buttyan 2001]; for example, it fails to capture information an adversary may gain from detailed knowledge of the protocols in use. An interesting topic for future research will be the adversary model in "open" ad hoc networks.

—It will be difficult to hold malicious nodes accountable for their actions, since they can always rejoin the network under a different (new) identity.

2.4. Limited Resources

Nodes have limited computational, memory, and energy resources in contrast to their wired predecessors. Nodes are small hand-held devices (possibly "off-the-shelf" consumer electronics) that do not hinder user mobility. In an attempt to keep the cost of these devices low, they are normally powered by a small CPU, accompanied by limited memory resources. As the devices are mobile, they are battery operated. This often results in short *on* times and the possibility of power failure due to battery exhaustion, perhaps during execution of a network-related function.

Devices may have limited bandwidth and transmission ranges. If it is assumed that advances in integrated circuit (IC) technology will keep on following Moore's law, computational and memory limitations will be alleviated in a matter of time. Bandwidth and transmission range (in the case of communication via radio transmissions) are unlikely to improve dramatically with respect to power consumption as both are dependent on Shannon's law and thus limited [Taub and Schilling 1991]. In order to achieve a higher bandwidth, a higher signal-to-noise ratio (SNR) is required, which in turn requires higher transmission power [Taub and Schilling 1991]. Higher transmission power significantly depletes battery power, which is unlikely to improve significantly given the current rate of advancement in battery technology [Ravi et al. 2004].

A security protocol that fails to optimize node and network resources will simply not be adopted in practice.

2.5. Poor Physical Security

Nodes are mobile and therefore cannot be locked up in a secure room or closet. These small hand-held devices are easily compromised by either being lost or stolen. It is therefore highly probable than an adversary can physically compromise one or more nodes and perform any number of tests and analysis. The adversary can also use the nodes to attack distributed network services, such as a distributed online certificate authority [Zhou and Haas 1999]. Poor physical security is not as relevant in "open" MANETs: the adversaries do not have to physically capture nodes to become an insider or to perform analysis on the protocols. The poor physical security of mobile devices may result in serious problems in "closed," military-type MANETs where physically compromised nodes can be used to launch active, insider attacks on the network.

2.6. Shared Physical Medium

The wireless communication medium is accessible to any entity with the appropriate equipment and adequate resources. Accordingly, access to the channel cannot be restricted. Adversaries are therefore able to eavesdrop on communications and inject bogus messages into the network without limitation. The shared channel and the nodes' poor physical security again emphasize that security mechanisms must be able to deal with the worst-case *active*, *insider* adversary.

2.7. Distributed System

Considering the above properties, nodes in ad hoc networks have a symmetric relationship. This implies that they are all equal and therefore should equally distribute all of the responsibilities of providing network functionality. This is not only for security reasons but to ensure reliable, available network services that place the same burden on the computational, memory and energy resources of all network participants [Zhou and Haas 1999; Capkun et al. 2003b]. It is anticipated that a fair distributions of services will also help to alleviate selfishness [Buttyan and Hubaux 2003].

3. APPLICATIONS OF MOBILE AD HOC NETWORKS

To understand the scope of MANETs and the usefulness of their unique characteristics, the potential applications of ad hoc networks are briefly considered.

Ad hoc networks have applications in two major fields: military and commercial environments.

3.1. Military Applications

The origin of networks that rely on no preexisting infrastructure can be traced back to the early 1970s with the DARPA and PRNET projects [Toh 2001; Haas et al. 2002], where the initial focus was on military applications.

The application of ad hoc networks in a military environment is particularly attractive because of their lack of infrastructure and self-organizing nature. Consider conventional networks that rely on infrastructure such as base stations: the infrastructure introduces points of vulnerability which may be attacked and, if eliminated, dismantle the operation of the entire network. In battlefield scenarios, robust and guaranteed communication is essential, with potentially fatal consequences if compromised. Ad hoc networks can continue to exist even in the event of nodes becoming disconnected due to poor wireless connectivity, nodes being compromised or switched off, nodes moving out of range, node being damaged during physical attack on users, or nodes failing due to malfunction or battery depletion.

Applications such as sensor networks [Akyildiz et al. 2002], positional communication systems [Quazi 2003; Dearham 2003], and tactical ad hoc networks [Jubin and Tornow 1987] will continue to be some of the driving forces behind ad hoc network development.

The main characteristic of military-type MANETs is the use of an offline authority. In *authority-based* MANETs, nodes share preestablished relationships initialized by the offline authority. The presence or absence of a priori security relationships has a fundamental impact on the design strategy of key management schemes for MANETs.

3.2. Commercial Applications

Commercial applications of ad hoc networks may include establishing connectivity in terrains where conventional networks, such as cellular networks, are not financially viable, cannot provide sufficient coverage, or need bypassing.

Private networks or personal area networks (for the purpose of teleconferencing, video conferencing, peer-to-peer communications, ad hoc meetings, or, more generally, collaborative applications of all kinds) are possible applications of ad hoc networks. It is anticipated that these applications will gain momentum as soon as the flexibility and convenience of self-organized ad hoc networking is fully appreciated and protocols are implemented with commercially available products. Take, for example, cellular networks: what was once seen as an impractical technology has now become a necessity.

Emergency situations caused by geopolitical instability, natural, or man-made disaster could result in existing networking infrastructure being damaged or becoming unreliable. For example, Hurricane Katrina struck New Orleans, Louisiana, on August 29, 2005. The storm destroyed most of the fixed communication infrastructure as it blanketed approximately 90,000 square miles of the Unites States, a region almost as large as the United Kingdom [FEM 2005]. In order to launch an effective disaster relief operation, communication is of the essence, even between a localized group of relief workers. "Open" MANETs will make it possible for relief workers from various countries to establish communication on the fly, therefore eliminating the time penalty in setting up and managing conventional, fixed-infrastructure networks. Search and rescue missions could also be conducted in locations not allowing access to existing communication networks. Search and rescue missions may also fall under the military applications category.

Vehicular ad hoc networks allow vehicles traveling along a highway to exchange data for traffic congestion monitoring, intervehicle communications, and early warning of potential dangers ahead such as an accident, road obstruction, or stationary vehicle.

Several research projects have been initiated to deal with vehicular ad hoc networking [Morris et al. 2000; Franz et al. 2001; Raya and Hubaux 2005].

4. KEY MANAGEMENT IN MOBILE AD HOC NETWORKS

As an introduction to key management, this article briefly considers the classification of security problems in MANETs. The aim is to position the problem of peer-to-peer key management within the MANET security field. The main observation is that cryptographic techniques are often at the center of solving security problems in MANETs and hence need key management [Zhou and Haas 1999; Hubaux et al. 2001].

Before introducing the different protocol groups, this section further clarifies what is meant by key management. The subsequent subsections also provide definitions and terminology for the different properties and requirements of key management schemes.

4.1. Motivation for Key Management in Mobile Ad Hoc Networks

Despite the evolution of MANETs over the past decade, there are still a number of security-related problems that are open [Zhou and Haas 1999; Hubaux et al. 2001]. This means that, although solutions have been proposed, none seems to satisfy all of the constraints of MANETs. Figure 1 illustrates the areas investigated within the MANET field, with particular focus on security issues. Note that this list highlights the main areas of ad hoc network security and could be expanded.

As illustrated in Figure 1, research in the MANET security field is concerned with a variety of different aspects. Researchers in the ad hoc network security field initially focused on secure routing protocols [Zhou and Haas 1999]. The focus of these protocols (Ariadne [Hu et al. 2002a], SEAD [Hu et al. 2002b], ARAN [Dahill et al. 2001], SRP [Papadimitratos and Haas 2002]) is twofold:

(1) To provide a routing mechanism that is robust against the dynamic network topology of MANETs.

(2) To provide a routing mechanism that offers protection against malicious nodes.

Routing protocols may use various security mechanisms to mitigate attacks on the routing infrastructure. Some of these mechanisms are redundancy exploitation; diversity coding; on-demand route discovery; route maintenance techniques; fault- or intrusion-tolerant mechanisms, and cryptographic mechanisms.

For example, routing schemes may exploit redundancy by establishing multiple routes from source to destination (as easily achieved by ZRP [Haas and Perlman 1998], DSR [Johnson and Maltz 1996], TORA [Park and Corson 1997], and AODV [Perkins and Belding-Royer 1999]) [Zhou and Haas 1999]. By sending data via all these routes, the redundancy will ensure that all data arrives at the destination. An alternative mechanism to sending data via redundant routes is *diversity coding* [Ayanoglu et al. 1993]. Diversity coding takes advantage of redundant routes in a more bandwidth-efficient way by not retransmitting the messages. Rather, it transmits limited redundant information through additional routes for the purpose of error detection and correction.

All of these mechanisms have various degrees of effectiveness. It is widely acknowledged that *cryptographic mechanisms* can provide some of the strongest techniques to ensure the availability, integrity, and confidentiality of routing information [Hubaux et al. 2001]. This observation also holds true for many of the other MANET security problems highlighted in Figure 1 [Hubaux et al. 2001]. If the basic networking mechanisms are considered, threat identification reveals that cryptographic techniques can also be used to mitigate attacks that exploit *over-the-air* communication, channel access mechanisms, and neighbor discovery [Hubaux et al. 2001].

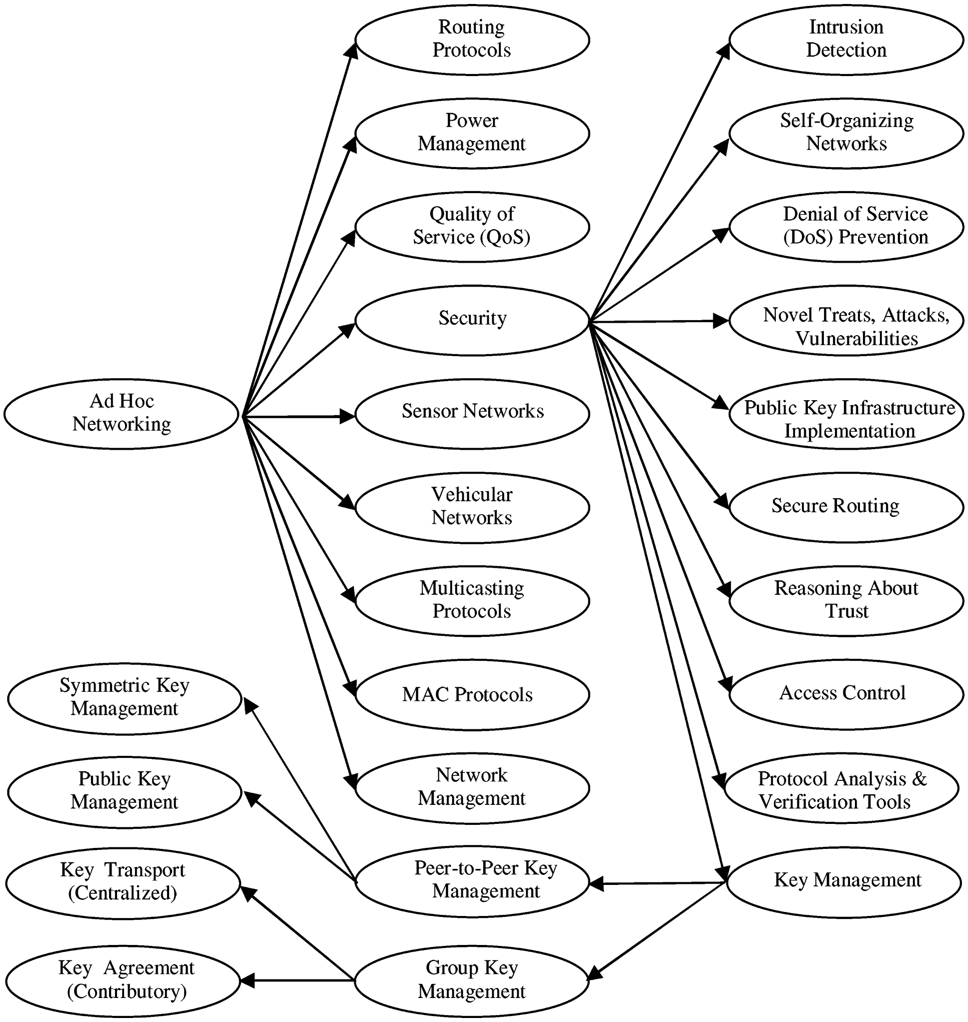


Fig. 1. Dimensions of ad hoc networks.

Secure key management with a high-availability feature is at the center of providing network security via cryptographic mechanisms [Menezes et al. 1996]. However, most routing schemes and related basic networking mechanisms neglect the crucial task of secure key management and assume preexistence and presharing of secret and/or private/public key pairs [Zhou and Haas 1999]. In fact, many cryptographic-based mechanisms that solve MANET security problems have a direct reliance on an efficient and secure key management infrastructure. This leaves key management techniques as an open research area in the ad hoc network security field [Zhou and Haas 1999; Hubaux et al. 2001].

4.2. Defining Key Management

A *keying relationship* is the state wherein network nodes share keying material for use in cryptographic mechanisms [Menezes et al. 1996]. The keying material can include public/private key pairs, secret keys, initialization parameters, and nonsecret parameters supporting key management in various instances. Key management can be defined as a set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties [Menezes et al. 1996]. In summary, key management integrates techniques and procedures to establish a service supporting [Menezes et al. 1996]:

(1) initialization of system users within a network;
(2) generation, distribution, and installation of keying material;
(3) control over the use of keying material;
(4) update, revocation, and destruction of keying material;
(5) storage, backup/recovery, and archival of keying material, and
(6) bootstrapping and maintenance of trust in keying material.

Authentication is the basis of secure communication. Without a robust authentication mechanism in place, the remaining security goals (confidentiality, data integrity, and nonrepudiation) are in most instances not achievable. Authentication can only be realized by means of verifying something known to be associated with an identity. In the electronic domain, the owner of the identity must have a publicly verifiable secret associated with its identity; otherwise, the node can be impersonated.

Authentication in general depends on the context of usage [Menezes et al. 1996]. Key management is concerned with the authenticity of the identities associated with the six services given above; it is a concept which may seem trivial at first, but one that is not easily achieved. Authentication of users is particularly difficult (and in most network settings impossible) without the help of a trusted authority.

The fundamental function of key management schemes is the establishment of keying material. *Key establishment* can be subdivided into key agreement and key transport [Menezes et al. 1996]. *Key agreement* allows two or more parties to derive shared keying material as a function of information contributed by, or associated with, each of the protocol participants, such that no party can predetermine the resulting value [Menezes et al. 1996]. In *key transport* protocols, one party creates or otherwise obtains keying material, and securely transfers it to the other party or parties [Menezes et al. 1996]. Both key agreement and key transport can be achieved using either symmetric or asymmetric techniques. A *hybrid* key establishment scheme makes use of both symmetric and asymmetric techniques in an attempt to exploit the advantages of both techniques [Menezes et al. 1996].

4.3. Requirements of Key Management Schemes

Key management services should adhere to the following generic security attributes:

—*Confidentiality*. Key management schemes should guarantee key secrecy, that is, ensure the inability of adversaries or unauthorized parties to learn keying material (or even partial keying material).

—*Key authentication*. Key authentication is a property whereby a communication entity is assured that only the specifically identified and authenticated communication entity may gain access to the cryptographic key material.

Key authentication, in the context of a communication session between two parties, can either be *unilateral* or *mutual*: unilateral authentication means that only one party's keying material is authenticated, while mutual authentication involves validating both parties' keying material.

Possession of the key is in fact independent of key authentication. Key authentication, without knowledge that the intended recipient actually has the relevant key, is referred to as *implicit* key authentication.

—*Key confirmation*. If key confirmation is provided by a key establishment protocol, communication entities prove possession of authenticated keying material. Key authentication with key confirmation yields *explicit* key authentication.

—*Key freshness*. The *key freshness* property improves security by ensuring new and independent keys between different communication sessions. By separating communication sessions, the available information for cryptanalytic purposes is limited, which makes cryptanalytic attack more difficult [Menezes et al. 1996].

—*Perfect forward secrecy*. *Perfect forward secrecy* (PFS) ensures that compromise of *long-term* keys cannot result in compromise of past session keys [Steiner et al. 2000; Ateniese et al. 1998; Menezes et al. 1996].

—*Resistant to known key attacks*. A key management scheme is vulnerable to *known key attacks* (KKA) if a compromised *past* session key or subset of past session keys allows [Steiner et al. 2000; Ateniese et al. 1998; Menezes et al. 1996] the following: (1) a *passive* adversary to compromise future session keys and (2) an *active* adversary to *impersonate* other protocol participants.

—*Forward secrecy*. A key management scheme with a forward secrecy property prevents an adversary from discovering subsequent keys from a compromised contiguous subset of old keys [Kim et al. 2000, 2004].

—*Backward secrecy*. A key management scheme with a backward secrecy property prevents an adversary from discovering preceding keys from a compromised contiguous subset of old keys [Kim et al. 2000, 2004].

—*Key independence*. Key independence guarantees that a passive adversary who knows a proper subset of keys cannot discover any other keys [Kim et al. 2000, 2004]. Key independence subsumes forward and backward secrecy. Key independence does not imply key freshness.

—*Availability*. A high-availability feature prevents degradation of key management services and ensures that keying material is provided to nodes in the network when expected.

—*Robustness*. The key management scheme should tolerate hardware and software failures, asymmetric and unidirectional links, and network fragmentation/partitioning due to limited/errorprone wireless connectivity [Sterbenz et al. 2002].

—*Survivability*. Survivability is the capability of the key management service to remain available even in the presence of threats and failures. Survivability goes beyond security and fault tolerance to focus on the delivery of services, even when the system is partly compromised or experiences failures. (Survivability thus subsumes robustness.) Rapid recovery of services is required when conditions improve [Sterbenz et al. 2002]. Survivability includes *byzantine robustness*, which implies that the key management service should be able to function properly even if some misbehaving participating nodes attempt to disrupt its operation.

More specifically, key management services with a survivability feature focus on the delivery of essential services (for example certification services in public key infrastructure) and the preservation of keying material (public key certificates, session keys, etc.). Survivability can be summarized by the *The Three Rs* [Sterbenz et al. 2002]:

—resistance: the capability of the system to defend against or tolerate attacks;

—recognition: the capability of the system to detect attacks in process and monitor the extent of the damage or compromise.

- —recovery: the main feature of survivability; it is the capability to maintain services during attack, limit the extent of the damage and restore full services following the attack.

—*Efficiency*. The key management service should be efficient with respect to communication, computational, memory, and energy resources.

—*Scalability*. Scalability ensures efficiency and availability as the number of networking nodes rapidly and significantly changes; the key management scheme should thus seamlessly scale to network size.

5. PEER-TO-PEER KEY MANAGEMENT FOR MOBILE AD HOC NETWORKS

As mentioned in Section 1, the focus of this article is on peer-to-peer key management for mobile ad hoc networks (MANETs). Investigations by the authors within the available publications have led to the classification of the current protocols into the following subsets:

(1) partially distributed certificate authority;
(2) fully distributed certificate authority
(3) identity-based key management;
(4) certificate chaining-based key management;
(5) cluster-based key management;
(6) predeployment-based key management;
(7) mobility-based key management, and
(8) parallel key management.

Most of the above subsets use public key cryptography due to its superiority in distributing keys, providing authentication, and achieving integrity and nonrepudiation [Zhou and Haas 1999; Menezes et al. 1996]. Symmetric key systems need a channel that provides both data integrity and confidentiality: the latter property may not always be readily available without any form of trusted authority or secure side channel (such as an infrared interface).

The *partially distributed certificate authority* group of protocols distributes the trust in the certificate authority to a subset of the network communication entities. The approach mitigates the single point of vulnerability inherent to the *centralized* certificate authority. Protocols considered to represent this implementation method were presented in Zhou and Haas [1999] and Yi and Kravets [2003], respectively (Section 6).

The *fully distributed certificate authority* protocol subset preserves the symmetric relationships between the communication entities in MANETs by distributing the burden of key management to *all* communication entities. Each authorized node in the network receives a share of the certificate authority's secret key, allowing neighbors to service requests for certification. The protocol that introduced this method was presented in Luo et al. [2002] (Section 7).

The *identity-based key management* approach borrows concepts from the *partially distributed certificate authority* protocols, but uses an identity-based cryptosystem to reduce the storage requirement compared to conventional public key cryptosystems. The protocol by Khalili et al. [2003] will be considered as representative of this protocol group (Section 8).

In the *certificate chaining-based key management* approach, communication entities can authenticate certificates by means of finding certificate chains between them. Certificate chaining can be explained by the following example: party *A* wants to communicate with party *C*, which requires party *A* to authenticate party *C*'s certificate.

The two parties have no communication history, but party A trusts the certificate of a third entity, party B. Party B informs party A that it trusts the certificate of party C. Party A that trusts party B will thus also trust party C as a result of party B's recommendation. There is thus a fully connected certificate chain between party A and C through party B, which enables party A to authenticate the certificate of party C without any previous communication. Section 9 investigates the protocol that introduced the certificate chaining based key management approach as detailed in Hubaux et al. [2001] and Capkun et al. [2003b].

The *cluster-based key management* subset relies on a clustering algorithm to subdivide the network into smaller groups. Group members in the same proximity can monitor their neighbors and make recommendations to members from other groups on the authenticity of their neighbors' certificates. The cluster-based subset is introduced by investigating the protocol presented in Ngai et al. [2004] (Section 10).

The *predeployment-based key management* subset makes use of an offline authority to issue each node with keying material prior to network formation. It is widely agreed that key predistribution techniques are ideally suited for establishing secure connectivity in large-scale distributed sensor networks [Eschenauer and Gligor 2002]. The limitations of sensor networks render conventional key establishment techniques (such as public key cryptography) unsuitable [Chan et al. 2003]. Section 11 introduces key predistribution techniques by giving an overview of the protocol in Eschenauer and Gligor [2002] and discusses subsequent improvements.

The *mobility-based key management* subset exploits mobility and node encounters to establish security associations and to warrant mutual authentication between users. In contrast to the previously discussed subsets, the protocols in this group introduce a shift in paradigm with respect to previous attempts to provide key management for *fully* self-organized MANETs. Rather than trying to adapt solutions suited for conventional wireline networks, the protocols in this subset use the unique characteristics of MANETs to their advantage. Section 12 investigates the symmetric and asymmetric key based protocols that introduced the mobility-based key management approach as presented in [Capkun et al. 2003a, 2006].

The combination of any of the above key management approaches gives rise to what the authors call the *parallel key management* subset. By using two or more of the above approaches in parallel, the advantages of the one scheme is used to mitigate the disadvantages of the other. This subset can be represented by the scheme introduced in Yi and Kravets [2004], which combines a *partially distributed certificate authority* [Yi and Kravets 2003] and the *certificate chaining-based key management approach* [Capkun et al. 2003b] (Section 13).

6. PARTIALLY DISTRIBUTED CERTIFICATE AUTHORITY APPROACHES

One of the first approaches to solve the key management problem in MANETs was published in Zhou and Haas [1999]. This approach was later extended in Yi and Kravets [2001, 2002a, 2002b, 2003]. Schemes are still actively proposed within this subset [Xu and Iftode 2004; Wu et al. 2005a, 2005b].

6.1. System Model

Zhou and Haas [1999] proposed a distributed public key management service for ad hoc networks, where the trust is distributed between a set of nodes by letting the nodes share the system secret. The distributed certificate authority (DCA), illustrated in Figure 2 [Zhou and Haas 1999], consists of n server nodes which, as a whole, have a public/private key pair K/k. The public key K is known to all nodes in the

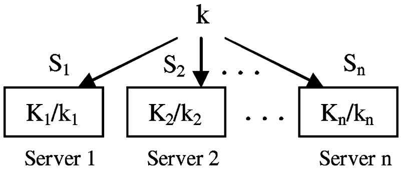


Fig. 2. Key management service K/k configuration.

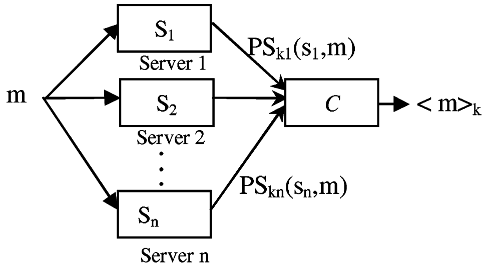


Fig. 3. Threshold signature K/k generation.

network, whereas the private key k is divided into n shares $(s_1, s_2, s_3, \ldots, s_n)$, one for each server.

The distributed certificate authority (DCA) signs a certificate by producing a *threshold group signature*, as shown in Figure 3 [Zhou and Haas 1999]. Each server generates a partial signature using its private key share and submits the partial signature to a combiner C. The combiner can be any server and requires at least $t + 1$ shares to successfully reconstruct the digital signature.

6.2. System Analysis

6.2.1. Initializatio Phase. The system as proposed in Zhou and Haas [1999] requires an *offline* trusted third party (TTP) to construct the distributed public key management service. Prior to network formation, the TTP uses a threshold secret sharing scheme [Shamir 1979] to generate shares $(s_1, s_2, s_3, \ldots, s_n)$ of the DCA's private key k. These shares are distributed to n arbitrary nodes (servers), which collectively form the DCA. The TTP must also issue all nodes in the network with the DCA's authentic public key K.

In order to prevent unauthorized nodes from getting certification services from the DCA, the offline authority will have to issue each node with a certificate signed by the private key of the DCA. If the certificate of each node is also stored on the DCA servers, preestablished security associations are available to authenticate certification requests. If the offline authority does not issue each node with its own certificate prior to network formation, the scheme is subject to the Sybil attack [Douceur 2002].

6.2.2. Certificate Retrieval. Nodes that require a certificate have to successfully contact at least $t + 1$ out of the n DCA servers. As illustrated in Figure 3, the threshold signature scheme proposed in Zhou and Haas [1999] makes use of a combiner node C to combine the partial digital signatures from the $t + 1$ servers. Any node can be chosen as a combiner, since no extra information about the private key k is disclosed to C. It is always possible for a combiner node to be compromised by an adversary or to be unavailable due to battery depletion or poor connectivity. As a solution, Zhou and Haas [1999] proposed selecting $t + 1$ nodes as combiners to ensure that at least one combiner can successfully reconstruct the digital signature. All nodes in the network, including the combiners, can verify the validity of the signature by using the CA public key, K.

The proposal presented in Yi and Kravets [2001, 2002a, 2003] differs from the original proposal in Zhou and Haas [1999], since the threshold signature scheme of Yi and Kravets does not require a combiner node C to construct the group signature. In Yi and Kravets [2001, 2002a, 2003], the DCA is called a *MObile Certificate Authority* (MOCA). In the MOCA framework, the communication pattern is one-to-many and visa versa, which means that a node that requires certificate services needs to contact at least $t + 1$ MOCA nodes and receive replies from each of them. The combining of the partial signatures is thus performed by the node requesting certification.

The original proposal in Zhou and Haas [1999] does not specify a communication protocol (certificate retrieval mechanism) for a node to contact the key management service. The proposal in Yi and Kravets [2001, 2002a, 2003] focuses primarily on the one-to-many-to-one communication pattern between a node and the MOCA. The MOCA certification protocol allows a node requiring certification services to broadcast *certification request* (CREQ) packets. Any MOCA node that receives the CREQ packet answers with a *certification reply* (CREP) containing its partial signature on the certificate. If the node successfully receives $t + 1$ valid CREPs in a fixed period of time, it can reconstruct the full certificate. If the certificate is verified to be correct, the certification request has succeeded. If the number of CREPs is insufficient after expiry of the node's CREQ timer, the process fails and the node can initiate another request. CREQ and CREP are similar to the *route request* (RREQ) and *route reply* (RREP) of on-demand ad hoc routing protocols (for example AODV [Perkins and Belding-Royer 1999] and DSR [Broch and Johnson 1999]). As CREQ packets are routed through nodes, a reverse path is established back to the sender. The reverse path is coupled with timers and maintained for a fixed time period to allow returning CREP packets to travel back to the node requesting the certificate service. In this case, an on-demand routing protocol and the MOCA certification protocol can benefit from each other by sharing routing information [Yi and Kravets 2003].

—*Flooding*. In the first implementation of the MOCA certification protocol presented in Yi and Kravets [2001], flooding is used for reliable data dissemination. As shown in simulation results presented in Yi and Kravets [2001], the flooding technique yields an effective approach to contact at least $t + 1$ servers, but generates high communication overhead. To prevent the potential broadcast storm nature of flooding, broadcast *ID*s are used (similarly to those in Perkins and Belding-Royer [1999]) such that all the CREQs generated by the same request are tagged with the same *ID* in order to allow intermediate nodes to drop requests that have already been forwarded.

—*β-Unicast*. To reduce the amount of certification traffic from flooding, while keeping an acceptable level of service, the method of β-unicast is introduced in the second report on the MOCA service [Yi and Kravets 2002a]. The method relies on multiple unicasts instead of flooding by using cached routing table information. The investigations in Yi and Kravets [2001] showed that a node client caches a moderate number of routes to MOCA nodes under reasonable certification traffic scenarios. The parameter β represents the threshold number of cached routes required by a node to use unicast instead of flooding. Flooding is thus the default method for contacting the MOCA in case the node fails to accumulate enough information in its cached routing tables. If the node perceives the network to be stable with relatively low mobility, $t + 1$ cached routes may be sufficient to initiate a multiple of unicast CREQs. To guarantee a reply from at least $t + 1$ servers, Yi and Kravets [2002a] introduced safety margin α. In the case of instability, the nodes should send out an additional α CREQs to increase the probability of successfully reaching $t + 1$ servers. The sum of α and the cryptothreshold t is called the *unicast threshold* and is represented by β. If there are more than sufficient (β) routes in the node's local cache, then the choice of which routes to use will affect performance. Three schemes were suggested in Yi and Kravets [2002a]:

—*Random MOCA nodes*. A random number β of MOCA nodes in the routing table are selected.

—*Closest MOCA nodes*. By utilizing the readily available hop count information in the routing table, β MOCA nodes with the minimum hop count are chosen.

—*Freshest MOCA nodes*. The most recently added β routes are used for β-unicast.

6.2.3. Certificate Revocation. Certificate revocation was not given much attention in Zhou and Haas [1999] or Yi and Kravets [2001, 2002a, 2003]. The authors proposed a simple certification revocation list (CRL) approach. In the MOCA framework, $t + 1$ nodes must agree to revoke a certificate. Each MOCA node generates a *revocation certificate* that contains information of the certificate to revoke. The MOCA node then broadcasts its partially signed revocation certificate across the network. Nodes that received $t + 1$ partial certificates will reconstruct the revocation certificate and update their local CRL.

6.2.4. Certificate Renewal. Expired certificates can be renewed by sending a CREQ message to any $t + 1$ MOCA nodes. Each MOCA node will update the certificate's contents with relevant information (for example, with a new expiry time) and newly bind the public key to the nodes identity by generating a partial signature.

6.2.5. Share Update. The key management services in Zhou and Haas [1999] and Yi and Kravets [2003] employ proactive key share refreshing [Herzberg et al. 1995] to thwart mobile adversaries and adapt to changes in the network. An adversary attempting to break the system must compromise more than the threshold t servers within the time interval between key updates.

6.3. Discussion and Comments on the Partially Distributed Certificate Authority Approaches

One of the advantages of the distributed certificate authority proposals is that they address the lack of server infrastructure in MANETs by distributing the functionality of a central authority among a group of users. The partly distributed system offers security services with a higher availability feature than a centralized server approach. The scheme, however, does not have a certificate revocation or synchronization mechanism to update servers.

The solution presented in Zhou and Haas [1999] has remnants of its wired predecessors, namely, a trusted authority, specialized server, and combiner nodes. The MOCA framework proposed in Yi and Kravets [2003], in contrast to the original DCA proposal [Zhou and Haas 1999], does not require a combiner server, C, effectively solving the problem of ensuring the availability of C.

The solution still has a largely centralized approach, although the threshold scheme allows t DCA servers to be compromised without sacrificing the key management service. One of the major assumptions of the solutions proposed in Zhou and Haas [1999] and Yi and Kravets [2003] is the presence of an *offline* TTP that initially empowers servers or distributes keying material before network formation. The assumption of an offline TTP makes the scheme unsuitable for *fully* self-organized MANETs. The information distributed by the offline TTP makes the solution nonscalable since all network certificates must be known a priori by the DCA servers in order to provide access control to certification services.

The communication overhead introduced by Zhou and Haas [1999] and Yi and Kravets [2003] is a point of concern as nodes need to contact the DCA every time they require certification. Any node can contact the DCA by flooding the network with a certification request. The flooding method is an effective way to contact t out of n DCA nodes [Yi and Kravets 2003]. The problem is that each of the DCA nodes respond with a certification reply causing a reverse packet storming effect of $\mathcal{O}(n)$ [Yi and Kravets 2003]. Yi and Kravets [2003] made a noteworthy effort to optimize the scheme by investigating different data dissemination techniques for contacting the DCA by means of simulation. Carter et al. [2003] investigated *manycast*, which appears to be a promising technique to contact only a subset of members from a group with minimum communication overhead. *Manycast* yields better performance than the flooding and β-unicast techniques discussed in Section 6.2, especially if *manycast* is integrated into the routing protocol.

Analysis by the authors on these schemes has shown that they are subject to the following weaknesses:

—The distribution of the system's private key is normally performed with a threshold secret sharing scheme [Shamir 1979]. Central to the security of the threshold cryptosystem is the choice of the security parameters (n, t), where n is the total number of nodes forming the DCA and t is the threshold of nodes that must be compromised to render the system insecure. The process of choosing appropriate parameters (n, t) is a very difficult task. The choice of these parameters inevitably forces a tradeoff between the security of the system and the availability of the DCA nodes.

Some of the factors to consider when choosing (n, t) are the networking environment of the MANET; physical security of the users' nodes; bandwidth requirement/utilization of the users; frequency of certification requests to the DCA; mobility patterns of users; capabilities of attackers, and the availability of the DCA nodes. For example, MANETs are subject to errorprone wireless connectivity, limited energy resources (poor battery life), and limited transmission ranges (see Section 2). Nodes forming the DCA may thus frequently be unavailable for the rendering of certification services. A hostile environment, such as those found in military applications, would require t to be set as high as possible. The problem, however, is that increasing t up to a "safe" point may prevent users from successfully contacting t out of n DCA nodes. The fundamental observation is that taking all these factors into consideration when choosing (n, t), while keeping in mind the security/availability tradeoff, is a difficult problem to solve and is central to the security provided by threshold cryptosystems. In the view of the authors, a strong security argument for MANETs is not possible with this approach.

—In threshold cryptosystems, it is impractical to assume that a *mobile* adversary cannot compromise more than t shareholders during the entire lifetime of the system [Zhou and Haas 1999; Herzberg et al. 1995; Luo et al. 2002]. This forces the nodes forming the DCA to execute a *share renewal protocol* [Herzberg et al. 1995] within a variable period T; the choice of T is influenced by similar factors to those that affect the selection of (n, t). The share renewal protocol has to be fully distributed as the DCA members have no access to a central online TTP.

The initial access structure $\Gamma_P^{(n,t)}$ of the share distribution scheme will not remain constant [Desmedt and Jajodia 1997]. Assuming the same shareholders to be present at all times is unrealistic and the security/availability tradeoff will also have to be altered (by reselecting (n, t) on the fly) as a function of system vulnerability, changing networking environment and current functionality of the cryptosystem. Users may randomly join or leave the DCA group; hence the DCA will exhibit *dynamic* group membership as associated with dynamic peer groups [Steiner et al. 2000]. The security parameter (n, t) will have to be adjusted to allow for the dynamic membership. This obligates the DCA nodes to execute a *secret redistribution protocol* [Desmedt and Jajodia 1997; Wong et al. 2002] in order to distribute the shares to a new access structure $\Gamma_{P'}^{(n',t')}$ on each membership change or security/availability tradeoff adjustment.

Analysis by the authors has shown that the overhead associated with fully distributed secret update and secret redistribution protocols may not be ideal or practical in MANETs. The authors' studies on existing, fully distributed secret update and secret redistribution protocols [Wong et al. 2002; Herzberg et al. 1995; Desmedt and Jajodia 1997] have shown that distributed share update/rekeying schemes have a high communication and computational cost, which will have to be executed more frequently in MANETs than expected. To put the overhead in more quantitative terms, (discrete logarithm-based) share updating requires $O(n+t)$ messages from each DCA member, while each DCA member performs $O(nt)$ exponentiations and generates $O(t)$ random numbers. The share renewal protocol has similar overhead with $O(n' + t')$ messages for each DCA member and $O(n't')$ exponentiations and $O(t')$ random numbers generated by each member of the new access structure $\Gamma_{P'}^{(n',k')}$. The network as a whole has a communication cost of $O(n^2 + nt)$ and $O(n'^2 + n't')$ for secret update and redistribution, respectively. In the analysis of the schemes, a synchronous broadcast system was assumed. It is generally agreed that network-wide synchronization is not easily achievable in MANETs. Without a synchronous broadcast system, it is not clear if fully distributed secret update and redistribution protocols are possible while defending against all known attacks [Fouque and Stern 2001; Zhang and Imai 2003].

To the best of the author's knowledge there is currently no secret sharing, secret update and secret redistribution schemes available that are suitable for MANETs.

- Nodes in MANETs have a symmetric relationship; hence nodes are all equal and therefore should fairly share or equally distribute the responsibility of providing all network functions. This is not only important for security reasons, but to enable the network to ensure reliable and available services that places the same burden on the computational, memory, and energy resources of all the network participants. The use of a DCA as online TTP violates the symmetric relationship between network nodes. When the DCA is formed by all the nodes in the network, as in Luo et al. [2002], the symmetric relationship is preserved. This, however, impairs the overall security of the system since an attacker can compromise *any t* nodes in the entire network to break the threshold cryptosystem (see Section 7).

  If there exists *heterogeneity* among network participants, the use of nodes with more advanced resources for the DCA nodes is not only unfair, but promotes selfishness or denial of service attacks. Unequally shared responsibility is also fundamentally against the notion of fully distributed systems and motivates localized areas of vulnerability. It is, however, noted that the notion of heterogeneity can be exploited in network settings where the nodes do not have symmetric relationships, such as those found in military-type networks [Yi and Kravets 2003]. In such a scenario, the burdened nodes will tolerate the exploitation for the benefit—or "survival"—of the network as a whole.
- Another issue that has also been surprisingly overlooked is how the members of the DCA will collaborate to sign certificates. This is a more difficult problem than one may think: designing group signatures is much more complex than designing single-party signatures. What is needed by the *distributed authority*-based schemes

proposed in Zhou and Haas [1999], Yi and Kravets [2003], and Luo et al. [2002] is a *threshold-multisignature* scheme [Li et al. 1994, 2001; Wang et al. 1998; Lee and Chang 1999]. *Threshold-multisignature* schemes can be differentiated from threshold *group* signature schemes [Pedersen 1997] by the fact that, by definition in the latter, the individual signers remain anonymous. In threshold *group* signature schemes, it is computationally difficult to derive the identities from the group signature with the exception of the group manager(s). In contrast, in *threshold-multisignature* schemes the individual signers are *publicly traceable* and do not enjoy anonymity. Consequently, the traceability property of threshold-multisignature schemes allows the individual signers to be held accountable in the public domain. The authors believe traceability of the individual signers is essential in MANETs. The current state-of-the-art threshold-multisignature schemes [Li et al. 1994, 2001; Wang et al. 1998; Lee and Chang 1999] are notoriously flawed [Michels and Horster 1996; Tseng and Jan 1999; Wang et al. 2003; Wu and Hsu 2004]. What is even more discouraging is that all the existing group signature schemes that have been proposed to date are for conventional networks. It is widely known that solutions suitable for MANETs, in most cases, require a shift in paradigm in order to mitigate the consequences of the unique characteristics of MANETs.

—The existing solution in the *partially distributed certificate authority* subset does not aim to break the routing-security interdependency cycle [Bobba et al. 2003].

In the view of the authors, a threshold cryptosystem may have other applications in MANETs, but are not ideal for realizing key management.

7. FULLY DISTRIBUTED CERTIFICATE AUTHORITY APPROACHES

In Kong et al. [2001] and Luo et al. [2002], a public key management solution was proposed based on the approach originally presented in Zhou and Haas [1999]. The solution also uses an $(n, k)^1$ threshold signature scheme to form a distributed certificate authority (DCA). The scheme enhances the availability feature of Zhou and Haas [1999] by choosing n to be *all* the nodes in the network. The private key *SK* of the DCA is thus shared among all the nodes in the network and enables a node requiring the service of the DCA to contact any k one-hop neighboring nodes. In contrast to Zhou and Haas [1999], no differentiation is made between server and client nodes with respect to certification services. The solution includes a share update mechanism to prevent more powerful adversaries from compromising the certification service. One of the latest proposals within the fully distributed subset can be found in Joshi et al. [2005].

7.1. System and Adversary Model

The network model considers an ad hoc wireless network with insecure, errorprone, and bandwidth-constrained communication channels. The network has no infrastructure and a dynamically changing topology. The following assumptions are made:

(1) Each node v_i has a unique identifier (*ID*) and is able to discover its one-hop neighbors.

(2) Each node holds a valid certificate signed by the DCA private key *SK* binding its *ID* to a public key P_{v_i}. A certificate signed by *SK* can be verified by the authentic public key of the DCA *PK*.

(3) One-hop communication is more reliable than multihop communication.

[1] In this section k will be used, instead of t, for the threshold parameter notation to be compatible with Luo et al. [2002].

(4) Each node has more than k one-hop neighbors at any time instance.

(5) Detection of node misbehavior is easier and more practical among one-hop neighbors in contrast to multihop nodes.

(6) Mobility of the network is characterized by the speed of the node with the highest speed S_{max}.

The proposal in Luo et al. [2002] and Kong et al. [2001] attempts to alleviate two types of attacks: denial of service (DoS) and adversary intrusion or node break-ins. Adversaries may issue DoS attacks on various layers of the network stack.

As defined in Herzberg et al. [1995], adversaries can be characterized by one of two models:

—*Model I.* During the entire network lifetime, an adversary cannot successfully attack and compromise more than k nodes.

—*Model II.* If the network's lifetime is divided into time slots T, an adversary cannot successfully attack and compromise more than k nodes within T.

Luo et al. [2002] and Kong et al. [2001] attempted to defend against Model II adversaries with a scalable parallel share update mechanism.

7.2. System Analysis

7.2.1. Initialization Phase of Localized Certification Service. The system as proposed in Luo et al. [2002] and Kong et al. [2001] requires an *offline* trusted third party (TTP). The RSA-based design has a system DCA with an RSA key pair $\{SK, PK\}$. *Prior* to network formation, the TTP distributes a certificate signed with the DCA private key *SK* to each node. The certificate is a binding between the nodes' unique *ID* and public key which may be verified with the DCA's authentic public key *PK* known by all nodes in the network. It should be clear that the localized certification service never creates or issues an initial certificate. Its functionality is the renewal of expired certificates or reissuing of certificates the users (or intrusion detection mechanisms) believe to be compromised by adversaries.

The offline TTP distributes the first k shares. In more accurate terms, the TTP distributes to the first k nodes in the network a polynomial share P_v of the certificate signing exponent, *SK* according to a random polynomial $f(x)$ such that $P_v = f(v)$. Upon completion of this task, the TTP has no further part in network operation.

The polynomial $f(x)$ can be defined as $f(x) = SK + \sum_{j=1}^{k-1} f_j x^j$, where $f_1, f_2, \ldots, f_{k-1}$ are uniformly distributed over a finite field F.

7.2.2. Localized Self-Initialization. Kong et al. [2001] proposed an algorithm used to distribute shares of *SK* to nodes joining the network. A node joining the network must have a certificate binding its *ID* and public key signed by *SK*. Only a joining node with a valid certificate, and hence verifiable with *PK*, can obtain a share of *SK*. Since their architecture is built on Shamir's [1979] threshold secret sharing scheme, only a coalition of k nodes can issue the uninitialized node with a share of *SK*.

The self-initialization protocol can be summarized in four steps:

(1) A joining node broadcasts to neighboring nodes a service request with additional local coalition information.

(2) Each neighbor (coalition member) selects a *nonce* (random number) for each other node if its *ID* is *lower* than the other node's *ID*. A complete *shuffling scheme* is used by the nodes to exchange these nonces between them. In the peer-to-peer exchange

of nonces, the nonces are negated by the node with the *lower ID*. The nonces are encrypted with the public key of the intended recipient coalition member.

(3) The encrypted shuffling packets or nonces are then routed to coalition members.

(4) Each of the coalition members computes a shuffled partial secret share *SS* from its polynomial share P_v in *SK*. Since it is possible to derive *SS* from P_v, *SS* is blinded to node v_x by adding the sum of all nonces to *SS*. After decrypting the nonces, each coalition member computes $SS' = SS + \sum nonces$ and transmits its computed partial share SS' to v_x.

7.2.3. Certificate Issuing and Renewal. As described in the *initialization phase*, each node holds a share of the private key *SK* according to the random polynomial $f(x)$. In Kong et al. [2001], node v_i first locates a coalition β of k neighbors $\{v_1, \ldots, v_k\}$ and broadcasts certification requests to the selected neighbors. Each node $v_j \in \beta$ consults its monitoring data and makes a decision to grant or refuse certification. If v_i is certified as legitimate, v_j returns a partial certificate P_{v_j} to the requesting node v_i. The node then recovers the certificate as a whole from the partial certificates using the *k-bounded coalition* offsetting algorithm as given in Kong et al. [2001]. This scheme remains functional even if coalition members are under attack since only k partial signatures are required from neighboring nodes.

In the later article [Luo et al. 2002], two drawbacks were identified with the approach in Kong et al. [2001]:

(1) If any of the nodes $v_j \in \beta$ fail during the requesting process, all the other partial certificates become useless.

(2) When v_j receives a request from v_i, the monitoring records may not provide sufficient information to grant certification. This will be the case in a network with high mobility where v_j and v_i have never met before or have had insufficient interaction.

In order to solve the first drawback, *dynamic coalescing* is introduced. This method stems from the observation that the coalition can be dynamically formulated from any responding nodes, instead of being specified a priori by the requesting node v_i. Figure 4 [Luo et al. 2002] illustrates how dynamic coalescing might overcome the first problem identified above.

To solve the second drawback and accommodate mobility, certification is granted if no bad records are found. The unavailability of records is thus taken as insufficient reason to deny certification.

7.2.4. Certificate Revocation. Certificate records maintained by node v_j consist of two components: monitored data (certificates and behavior) of neighboring nodes and a certificate revocation list (CRL). The CRL is a list containing user *ID*s and accusers. If a node v_j concludes by direct data monitoring that a neighboring node is compromised, it marks the node "convicted" in its own CRL. The accuser v_j also floods the network with a signed accusation against the node. Now assume node v_j receives a signed accusation against an accused node, it checks in the CRL if the node has been previously marked "convicted." If this is the case, the message is regarded as being a confirmation of the conviction and dropped; if not, the node is marked as "suspect." To avoid the conviction of legitimate nodes, at least k accusations against a node is required before it is marked "convicted."

7.2.5. Parallel Share Updates. For a share distribution system to be robust against Model II adversaries, periodic updates are required [Herzberg et al. 1995]. This prevents an attacker from compromising more than k secret shares in the period

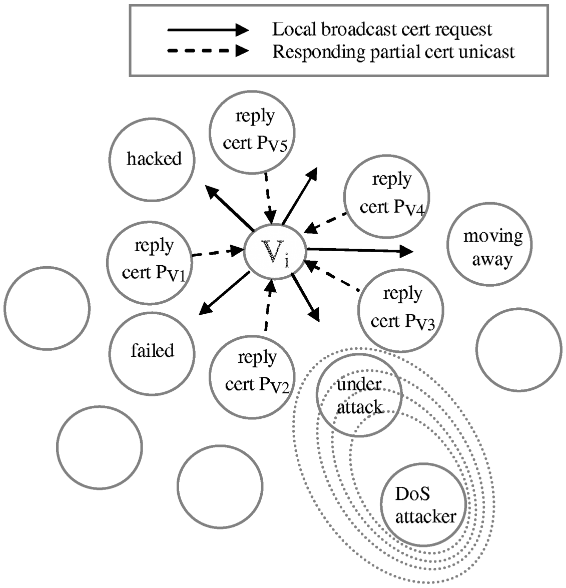


Fig. 4. Dynamic coalescing.

between periodic share updates. The proposal in Kong et al. [2001] gives two approaches to achieve share updates. The first approach is a process based on *localized self-initialization* explained in Section 7.2.2. The second approach features parallel share updates with faster convergence. The update is performed by distribution of a new random polynomial $f_{UPDATE}(x)$ whose coefficients are encrypted with *SK* to ensure authenticity. The node's new share in *SK* can then be collaboratively evaluated as $P_{vUPDATE} = f_{UPDATE}(v_i)$ by k neighboring nodes. Each neighbor returns its partial share update to v_i in a manner similar to that of the certification service. The parallel share update approach thus consists of three steps: collaborative generation of the update polynomial $f_{UPDATE}(x)$; robust propagation of the update polynomial's coefficients, and distributed evaluation of share updates.

7.3. Discussion and Comments on Fully Distributed Certification Authority Approaches

The fully distributed certificate authority proposal represents an improvement on the original partially distributed CA scheme in Zhou and Haas [1999] by fairly distributing the burden of holding a share of the secret key to all nodes in the network. This effectively preserves the symmetric relationship between network participants. The availability of the key management service is increased as now any k nodes, in a local neighborhood, can renew or issue a certificate. Nodes not in possession of a share can also contact at least k nodes to obtain a share.

Although innovative in design, the proposal suffers from most of the weaknesses discussed in Section 6.3. As in the original proposal [Zhou and Haas 1999], a trustworthy *offline* authority must issue every node before network formation with a certificate, binding a unique *ID* to a public key. This requirement makes the proposal unsuitable for fully self-organized ad hoc networks. Similarly to Zhou and Haas [1999] and Yi and Kravets [2003], this solution is also nonscalable since all identities must be known a priori.

Since the number k is a tradeoff between security and availability, the solution requires a way of adjusting k as the network expands. As pointed out in Capkun et al. [2003b], it is not clear how k will be adjusted in a network with a rapidly increasing or decreasing node density. The increase in availability of the certification service also comes at the cost of security since *any* k nodes in the network can be compromised to break the system; the authors do not believe that the assumption of a Model II adversary (Section 7.1) is realistic if n spans the entire network. For this reason k must always be chosen large enough to ensure the security of the system.

The assumption that each node will always have at least k one-hop neighbors is limiting: nodes may find themselves with less than k neighbors more frequently than expected.

As in the case of Zhou and Haas [1999] and Yi and Kravets [2003], it is not clear if this approach can break the routing-security interdependency cycle [Bobba et al. 2003].

8. IDENTITY-BASED KEY MANAGEMENT APPROACHES

ID-based cryptography [Joye and Yen 1998; Boneh and Franklin [2001]; Cha and Cheon [2003] originated from the need to reduce the memory storage cost of conventional public key systems and the burden of obtaining explicitly authentic public keys. Public keys in an *ID*-based scheme are nothing other than the identities of the users themselves. The identities, which are publicly known data, must uniquely identify the users. *ID*-based schemes thus uniquely bind private keys to identities.

The identity-based signature schemes are normally specified by four randomized algorithms [Boneh and Franklin 2001]:

(1) *Setup*. The setup algorithm takes as input security parameters and returns a master public/private key pair K_M/k_M for the system. The master private key is only known by the trusted third party (TTP) or private key generator (PKG) of the system.

(2) *Extract*. The extract algorithm takes as input the master private key and an identity *ID* and returns the personal private key corresponding to the *ID*.

(3) *Encrypt*. The encrypt algorithm takes as input the master public key K_M, the *ID* of the recipient, and a message m and returns the corresponding ciphertext. Note that *ID* serves as the public key of the recipient.

(4) *Decrypt*. The decrypt algorithm takes as input the master public key, a ciphertext, and the personal private key and returns the original message encrypted with the *ID* corresponding to the personal private key.

The personal private keys in an identity-based cryptosystem can also be seen as an *implicit symmetric key certificate*, that is, the personal private key is encrypted with the master private key of the PKG.

8.1. System Model

In Khalili et al. [2003], identity-based cryptography [Joye and Yen 1998; Boneh and Franklin 2001] is combined with threshold cryptography [Menezes et al. 1996] to avoid the “extensive” computational cost of public key cryptography. This solution is similar to that in Zhou and Haas [1999] with the threshold certificate authority replaced by a *threshold* private key generator (PKG). The randomized algorithms discussed above collectively realize an identity-based cryptosystem. All these algorithms will have to be adapted to a threshold cryptosystem and implemented on the nodes forming the distributed authority, that is, the PKG. How to modify the existing identity-based cryptosystem algorithms to suit a threshold cryptosystem is not discussed in Khalili et al.

[2003]. Single-party *encrypt* and *decrypt* algorithms will have to be implemented on all nodes forming the network.

8.2. System Analysis

8.2.1. Initialization Phase. On network formation, users agree on a key issuing policy and exchange all relevant security parameters. These must be mutually acceptable and nodes not approving of these parameters may choose to abort the network formation process. The initial set of nodes then form a *threshold private key generation service* (PKG), which generates a master public/private key in a distributed manner. The master private key is thus distributed to n nodes, each holding a share of SK^*. An adversary with less than threshold t shares cannot recover the master private key. The master public key in turn is given to all joining members of the network.

8.2.2. Registration Phase. After the initialization phase, the PKG can start issuing users with their private keys based on their identity. Khalili et al. [2003] proposed that nodes use their medium access control (MAC) or network layer address as an identity when contacting the distributed authority for their personal private key. The node contacts at least t of the PKG servers, forming the PKG, which each reply with their part of the requesting node's private key. Upon receipt of t correct shares, the user can compute its private key. Khalili et al. [2003] argued that nodes that are unable to contact t or more servers can roam the network in search of more shares.

8.3. Discussion and Comments on Identity-Based Key Management Approaches

In the initial phase of the proposed scheme by Khalili et al. [2003], nodes decide on a mutual set of security parameters. Any node that is not satisfied with the choice of parameters can choose not to participate in the network. Khalili et al. [2003] stated that their scheme is independent of the initial negotiations. This independence is difficult to see as it is the responsibility of the key management protocol to successfully initialize all system users within a domain [Menezes et al. 1996] (see Section 4.2). If adversaries are able to influence the selection of system parameters, they will be able to force nodes not to participate in the network.

The proposal does not address the issue of how the initial set of nodes will form the PKG or how a node will obtain a private key from the *distributed* PKG. Distribution of the master private key, as mentioned in Khalili et al. [2003], can be done using the algorithm presented in Gennaro et al. [1999]. Integration of the distributed key generation scheme [Gennaro et al. 1999] into the *setup algorithm* of the identity-based signature scheme [Cha and Cheon 2003] will enable the generation of a distributed master private key. A problem is noted with the implementation of *extract algorithms* of existing identity-based signature schemes in distributed systems. It is noted that the *extract algorithms* in Cha and Cheon [2003], Boneh and Franklin [2001], and Joye and Yen [1998] are designed for an entity obtaining a personal private key from a *centralized* PKG. Any centralized service in ad hoc networks is a single point of vulnerability. The *extracting algorithms* will have to be modified for negotiation of a private key with a *distributed* PKG. See observations in Section 6.3 on secret sharing, secret update, secret redistribution, and threshold-multisignature schemes.

The proposal does not avoid the weaknesses of *ID*-based cryptography. The major problem with *ID*-based cryptographic schemes is that they yield only *level* 1 trust [Petersen and Horster 1997], that is, the private key of users is known by the trusted authority. In conventional networks, this is not so much of a problem, but in MANETs

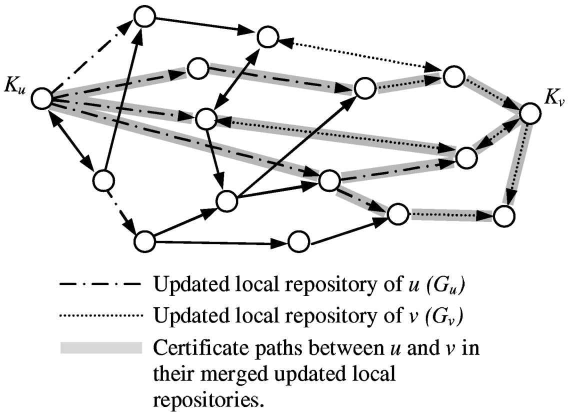

K_u
K_v
Updated local repository of u (G_u)
Updated local repository of v (G_v)
Certificate paths between u and v in their merged updated local repositories.

Fig. 5. A certificate graph and certificate paths between users u and v in their merged updated local repositories.

where the trusted authority is distributed between *online* servers or emulated by an arbitrary *offline* entity, this may not be feasible.

Assume the node can negotiate a personal private key with a *distributed* PKG; a major problem is how the PKG will securely transfer to the requesting node its personal private key shares. In the scheme proposed in Khalili et al. [2003], the requesting node shares no secret with the PKG, for example, a common symmetric key, nor do the nodes have public/private key pairs. It is therefore not clear how a node will obtain its personal key from the PKG in the presence of an adversary. This problem can only be solved by setting up some secure channel or by predistributing common keying material, neither of which is ideal in ad hoc networks.

The proposal swaps one difficult problem for another. In the case of a PKI solution, the primary concern is the authentication of public keys. In Khalili et al. [2003], the problem is to authenticate the identity of a node before sending the shares of the personal private key corresponding to the identity.

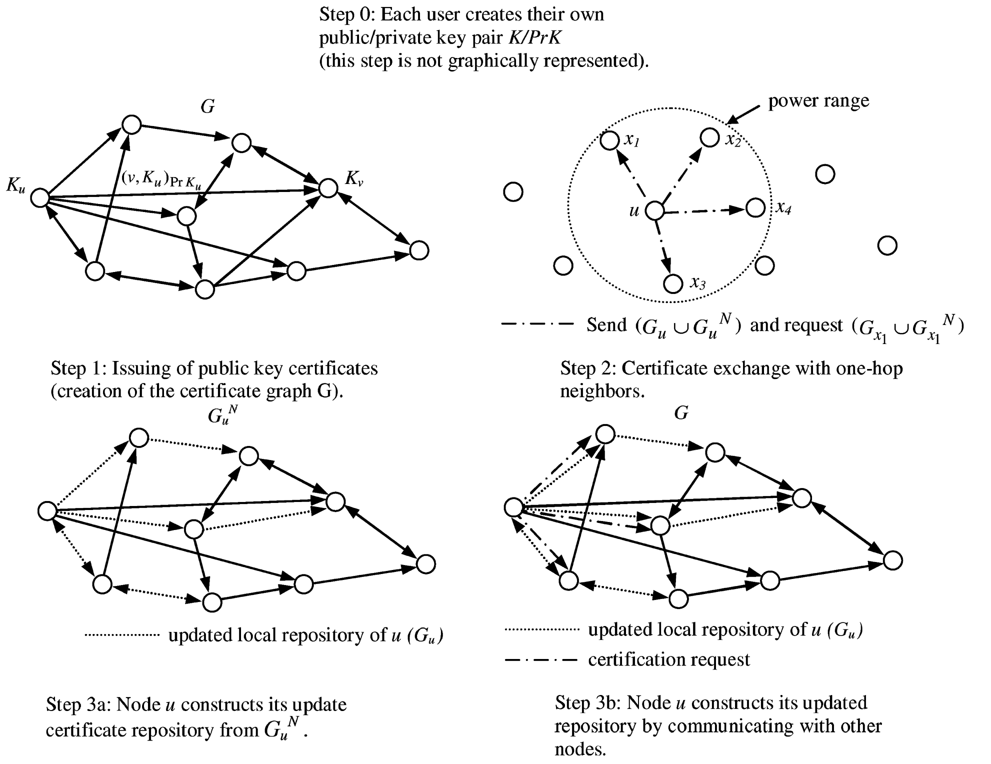
The solution is vulnerable to a *man-in-the-middle* attack on nodes joining the network [Khalili et al. 2003].

This approach also cannot break the routing-security interdependency cycle [Bobba et al. 2003].

Another example of a protocol within the identity-based key management subset can be found in Deng et al. [2004].

9. CERTIFICATE CHAINING-BASED APPROACHES

One of the most recent proposals presented in Capkun et al. [2003b] takes a step closer to meeting the constraints of MANETs. Unlike previous solutions, the public key infrastructure (PKI) in this proposal does not require *any* trusted third party. This makes the scheme suitable for fully self-organized MANETs. Each node issues its own certificates to other nodes in a manner similar to Pretty Good Privacy (PGP) [Zimmermann 1995]. It differs from PGP by the fact that there are no centrally managed certificate directories (online certificate servers), but certificates are rather stored and distributed by nodes in a self-organized nature. Each node keeps a limited certificate repository comprised of certificated nodes in its local neighborhood. As illustrated in Figure 5 [Capkun et al. 2003b], when a node u wants to validate the certificate of another node v, the nodes combine their certificate repositories and u attempts to find a chain of valid public key certificates between them.



Step 0: Each user creates their own public/private key pair *K/PrK* (this step is not graphically represented).

Step 1: Issuing of public key certificates (creation of the certificate graph G).

Step 2: Certificate exchange with one-hop neighbors.

Step 3a: Node *u* constructs its update certificate repository from G_u^N.

Step 3b: Node *u* constructs its updated repository by communicating with other nodes.

G
K_u
$(v, K_u)_{\mathrm{Pr}K_u}$
K_v
power range
x_1
x_2
u
x_4
x_3
Send $(G_u \cup G_u^N)$ and request $(G_{x_1} \cup G_{x_1}^N)$
G_u^N
updated local repository of u (G_u)
G
updated local repository of u (G_u)
certification request

Fig. 6. Four steps in initial phase of certificate chaining proposal.

9.1. System Model

Capkun et al. [2003b] presented their scheme in terms of an abstract model. In their model, the public keys and the certificates of the system are represented as a directed certificate graph $G(V, E)$, where V and E stand for the set of vertices and the set of edges, respectively (Figure 5). The vertices of the certificate graph represent public keys and the edges represent certificates. More precisely, there is a directed edge from vertex K_u to vertex K_w if there is a certificate signed with the private key of u that binds K_w to an identity. A certificate chain from a public key K_u to another public key K_v is represented by a directed path from vertex K_u to vertex K_v in G. Thus the existence of a certificate chain from K_u to K_v means that vertex K_v is reachable from vertex K_u in G (denoted by $K_u \longrightarrow_G K_v$).

9.2. System Analysis

9.2.1. Initialization Phase. The initial phase of the system is executed in four steps: each node creates a public/private key pair; each node creates a self-certificate, issues certificates to other nodes, and constructs an nonupdated certificate graph; nodes exchange certificates, and they create updated certificate repositories. Each of these steps is illustrated in Figure 6 [Capkun et al. 2003b] and explained in more detail in the sections that follow. Note that the step numbering is kept consistent with the numbering used in Capkun et al. [2003b].

9.2.2. Step 0. Creation of Public/Private Key Pairs. Similar to PGP [Zimmermann 1995], users locally create their own private key and corresponding public key.

9.2.3. Step 1. Creation of Self-Certificates and Issuing of Public Key Certificates. Individual public key certificates are issued by the users themselves with a limited validity period. When a node u has confidence in the public key/identity binding of node v, node u can issue a certificate (recommendation) to vouch for the binding. The nodes use these certificates to start constructing a certificate graph G.

9.2.4. Step 2. Certificate Exchange. The certificate exchange mechanism allows users to share and distribute certificates in their repositories. Certificates are stored at least twice: by the issuer of the certificate and by the user to whom the certificate is issued. The certificate exchange process consists of the periodic exchange of certificates between nodes and their neighbors. Exchanges are asynchronous. Users send their updated subgraphs G_u and nonupdated subgraphs G_u^N to their neighbors, who use the certificates to create or expand their nonupdated subgraphs. The message contains only the hash value of the certificates. The node checks the hash values against those held in its repositories and requests only the certificates with a negative hash-value comparison. The certificate exchange process has a low communication cost since certificate exchanges are only performed locally in a one-hop fashion.

9.2.5. Step 3. Construction of Updated Certificate Repositories. The nonupdated repositories (subgraphs) provide the nodes with only an incomplete view of the certificate graph. An updated certificate repository is constructed by node u by selecting a subgraph G_u of G. This can be performed in two ways: nodes can use the same local repository construction algorithm to explore only a relevant part of the certificate graph G (Step-3a, Figure 6) or construct an updated repository by communicating with their certificate graph neighbors (Step-3b, Figure 6).

9.2.6. Certificate Revocation. Users can revoke any issued certificate to other users if they lose their trust in the public key/identity binding. Similarly users can also revoke their own certificate if they believe that their private key has been compromised. Capkun et al. [2003b] proposed two revocation schemes: *explicit* and *implicit*.

In the *explicit* scheme the user issues an explicit revocation message to nodes in its local neighborhood. This will most probably be the users that usually request certificates from the revocation node under normal operation.

The *implicit* revocation scheme is based on the expiration time of the certificates. The scheme assumes that users will establish communication and exchange an updated version of the certificate within the valid period.

9.3. Discussion on Certificate Chaining-Based Approaches

The *fully* self-organized key management scheme, presented in Hubaux et al. [2001] and Capkun et al. [2003b], was the first to give an indication that key management, without any form of TTP, may be possible in MANETs. The self-organized scheme is clearly an advance on previous efforts [Zhou and Haas 1999; Yi and Kravets 2003; Luo et al. 2002; Khalili et al. 2003; Ngai et al. 2004] to provide key management for MANETs: eliminating any form of online TTP eliminates most of the problems associated with these schemes (see Section 6.3 for a discussion on the distributed certificate authority proposals). The *fully* self-organized scheme presented in Hubaux et al. [2001] and Capkun et al. [2003b] was designed for "open" MANETs and therefore not truly comparable with the schemes that assume an offline trusted authority.

Hubaux et al. [2001] and Capkun et al. [2003b] addressed a very difficult problem. How does one explicitly authenticate the public key of a user without any form of offline or online trusted authority? In MANET, this problem has to be solved with sporadic connectivity, while optimizing communication and computational resources. The problem can alternatively be defined in terms of trust establishment. If a user *A* trusts the certificate of another user *B*, user *A* has confidence that the public key contained in the certificate belongs to user *B*. Users *A* and *B* therefore have a *direct* trust relationship. Taking this notion a step further, user *A* can support the authenticity of the certificate of user *B* by signing the certificate with its own private key; any other user in the network that trusts the certificate of user *A* will also trust the certificate of user *B* if they are able to verify the recommendation of user *A*, and hence can verify the signature. If user *A* also recommends the certificates of other users, a *hierarchical* trust model is created. Certificate chaining used in PGP [Zimmermann 1995] naturally evolves from a direct and hierarchical trust model combination.

Hubaux et al. [2001] and Capkun et al. [2003b] successfully adapted the certificate chaining authentication approach to MANETs. The main difference between their scheme and PGP [Zimmermann 1995] is that the latter stores certificates in centralized repositories. In the Hubaux et al. [2001] and Capkun et al. [2003b] scheme, the certificates are disseminated and stored by all nodes without any assistance from a trusted authority.

Trust relationships take time to form and require user interaction. The *fully* self-organized scheme [Hubaux et al. 2001; Capkun et al. 2003b] therefore introduces a delay in the setup of security associations. As a result, the solution may encounter a problem in the initial phase when the number of issued certificates is insufficient to yield a sufficiently dense certificate graph.

Inherently a chain of trust provides *weak* authentication [Christianson 1996; Abdul-Rahman and Hailes 1997]. A common assumption in most distributed authentication protocols is that trust is implicitly transitive [Abdul-Rahman and Hailes 1997]. Trust transitivity means, for example, that if Alice trusts Bob who trusts Clark then Alice will also trust Clark. This has been shown to be *generally* untrue [Christianson 1996]. Josang et al. [2003] defined a valid transitive trust chain as a chain where every link in the chain contains the same trust purpose. Abdul-Rahman and Hailes [1997] pointed out that valid transitive trust chains satisfy four conditions with reference to the example above [Abdul-Rahman and Hailes 1997]. It is concluded from Christianson [1996], Abdul-Rahman and Hailes 1997, and Josang et al. [2003] that it is very difficult to ensure valid transitive trust chains with more that two links. The authors felt that users (in general) are not able to make intuitive security related decisions. The public's lack of even the most basic knowledge (such as what is a public key certificate) makes any scheme that relies on users *reasoning* about security vulnerable to attack. Examples of schemes that avoid such "user reasoning" have been presented in Capkun et al. [2006], Cagalj et al. [2006], and McCune et al. [2005]. In a two-link trust chain, Alice has a direct trust relationship with Bob and relies on recommendations from Bob based on this direct trust relationship. In practice a direct trust relationship implies that Alice and Bob know each other personally. This means that Alice can check with Bob if they share the same trust purpose or trust conditions. If the chain is extended to three links (four users), then Alice will have an indirect trust relationship with Clark, and hence may not know him personally. The indirect trust relationship lessens Alice's ability to ensure that she and Clark have the same trust conditions.

Furthermore, a chain is as strong as its weakest link. If any node along the chain is compromised or subject to byzantine behavior, it may result in false authentication. In "open" or *fully* self-organized networks, this may be even more relevant than in "closed" networks since an adversary does not have to compromise nodes to participate in the certificate exchange mechanisms.

10. CLUSTER-BASED KEY MANAGEMENT APPROACHES

The key management scheme proposed in Ngai et al. [2004] originates from the certificate chaining approach [Capkun et al. 2003b]. The authors assumed a cluster-based network model constructed with the *zonal algorithm* [Chen and Liestman 2003]. The zonal algorithm for clustering ad hoc networks partitions the network into different subsets using a distributed algorithm for finding the minimum spanning tree (MST). Once the network is partitioned and the MST determined for each subset, the algorithm computes the weakly connected dominating sets of the regions. Finally, it fixes the borders of the clusters, that is, connects unjoined regions, by the inclusion of additional nodes in the sets. Nodes clustered together in the same region form a group and are assigned a unique *ID*. The nodes learn the group *ID*s of other nodes by exchanging messages.

10.1. System/Trust Model

Each user is responsible for the creation of their own public/private key pair and generation of a self-certificate. Any node can sign the public key certificate of another node in the same group upon request. Nodes are assumed to have some monitoring components that enable them to observe the other nodes' behavior in their group and assign each node a trust value. The trust value is defined as an authentication metric, which represents the assurance with which a requesting node s can obtain the correct public key of a target node t. The trust between nodes in the same group is referred to as *direct* trust. The trust relationship between nodes in different groups are referred to as *recommendation* trust. The trust model is shown in Figure 7 [Ngai et al. 2004]. Each node should thus have a *trust table* for storing the trust values and related public keys of nodes it "knows" in the network.

10.2. Public Key Certification and Trust Value Update

If a node s wants to obtain the public key certificate of some other node t, s performs the following procedure [Ngai et al. 2004]:

(1) Node s looks up the group *ID* of t, denoted as φ_t.

(2) Node s consults its trust table and sorts the trust values of the nodes known to s in group φ_t. Let $i_1, \ldots, i_n \in I$, where i_n denotes the node with the highest trust value.

(3) Node s then sends certification request messages to each node in subset I. These nodes are also referred to as the *introducers*.

(4) Node s collects the reply messages $m \in M$ from I where $m = \{P_{k_t}, V_{i_k,t}, \ldots\}Sk_{i_k}$. P_{k_t} denotes the public key of t, $V_{i_k,t}$ denotes the trust value from i_k to t. Sk_{i_k} denotes the secret key of i_k which is used to generate a signature on m.

(5) Node s compares the public keys received from I and follows the majority votes. Let $i_{good} \in I_{good}$ and $i_{bad} \in I_{bad}$, where I_{good} are the nodes thought to be honest and I_{bad} the remaining perceived dishonest nodes. Node s perceives the public key of t received from I_{good} to be authentic.

(6) The trust values of the nodes $\in I_{bad}$ are reduced to zero. Node s then computes and updates the trust value of t using the following equation, where i_k denotes the

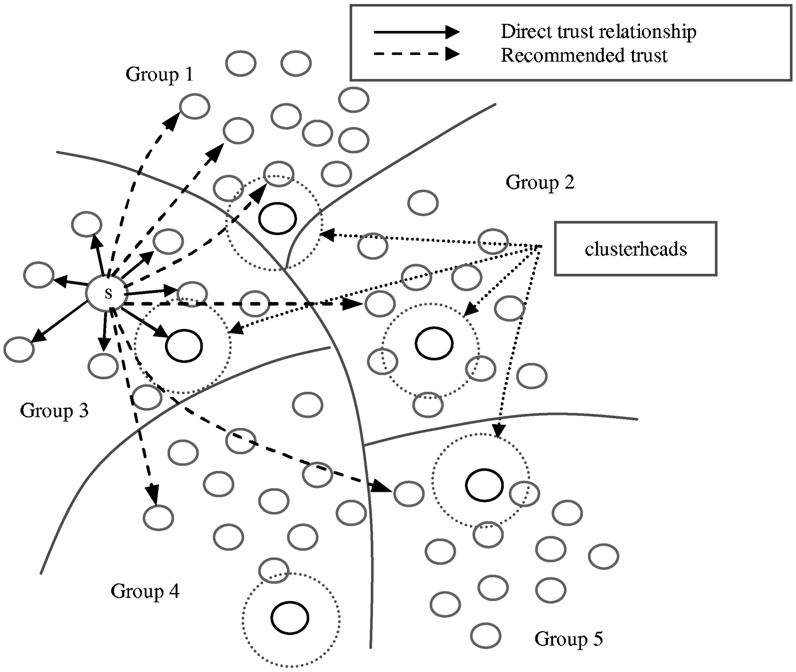


Fig. 7. Cluster-based trust model.

nodes and n the number of nodes in I_{good}:

$$V_{s,i_k,t} = V_{s,i_k} \bigodot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}} \tag{1}$$

and

$$V_t = 1 - \prod_{k=1}^{n}(1 - V_{s,i_k,t}). \tag{2}$$

The public key certification and trust value update procedures are illustrated in Figure 8 [Ngai et al. 2004] and Figure 9 [Ngai et al. 2004], respectively.

Step 6 may need some further explanation. After receiving, decrypting and comparing the trust values of I in step 5, s can calculate the new recommendation trust relationships from s to t via the nodes in $i_k \in I$ using Equation (1). Note that the nodes $i_{bad} \in I_{bad}$ make no contribution since their trust values are reduced to zero. The $\bigodot$ operator is defined in Beth et al. [1994] and is given as $V_1 \bigodot V_2 = 1 - (1 - V_2)^{V_1}$. In Beth et al. [1994], derivatives of the formula $V_1 \bigodot V_2$ were used to compute new trust relationships between V_1 and V_2 based on the direct trust values and recommendation trust values between them.

Once $V_{s,i_k,t}\ \forall\, i_k$, has been computed, s can compute the ultimate trust value V_t of t as seen by s after pubic key certification using Equation (2).

10.3. Discussion on Cluster-Based Key Management Approaches

This discussion will refrain from a detailed evaluation of network clustering in MANETs, but will rather focus on some issues related to the analysis of cluster-based key management schemes.

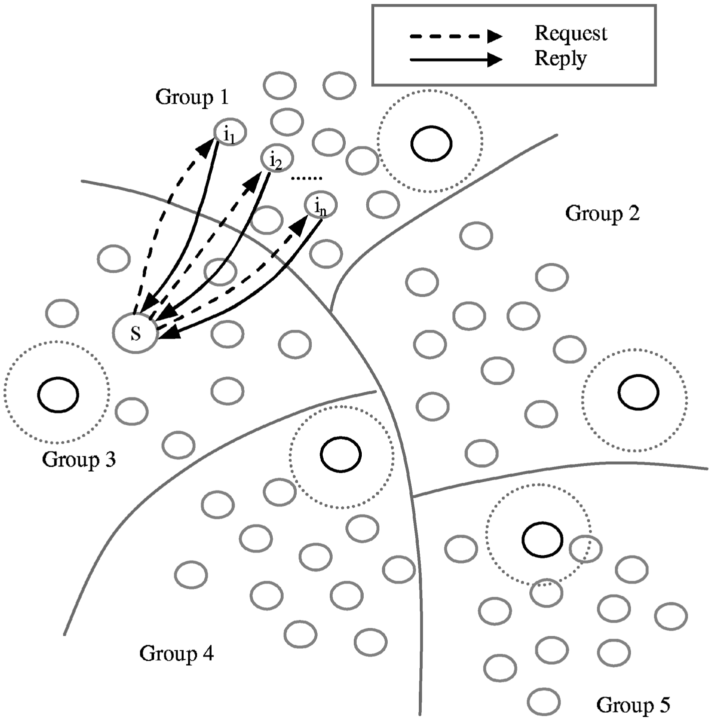


Fig. 8. Public key certification.

- When MANETs scale, route calculations become increasingly expensive [Chen and Liestman 2003]. The task of routing algorithms may be simplified by confining route discovery to a substructure of the network. By introducing clustering into the network, *local* messages can be transmitted on short paths within the same cluster, while *long-distance* messages travel longer distances from cluster to cluster. Each cluster is assigned a *clusterhead* as representative of the cluster. The clusterhead takes on the responsibility of participating in intercluster route calculation and long-distance message forwarding. The long-distance message forwarding requires the clusterheads to use more transmission power, which will significantly contribute to the depletion of the node's energy resources [Haas and Tabrizi 1998].
- The dynamic nature of MANETs makes clustering very problematic. The main concern is the selection, configuration, maintenance, and replacement of clusterheads. The role of a clusterhead should be able to be performed by any node in the network. Clusterheads are therefore no different from other nodes in MANETs and thus exhibit some similar characteristics. Due to unreliable connectivity and route failures, clusters may also partition. The frequent maintenance of clusterheads and cluster membership is unavoidable and may therefore cause impractical computational and communication overhead.
- The assignment of clusterheads is very difficult since nodes may not voluntarily take on this responsibility [Buttyan and Hubaux 2003] or necessarily have the required capacity to accommodate the additional overhead.
- The clusterheads become very convenient central points of attack for adversaries, for example, an adversary which can establish itself as a clusterhead effectively controls the whole cluster. Clearly this will not work in *fully* self-organized MANETs.

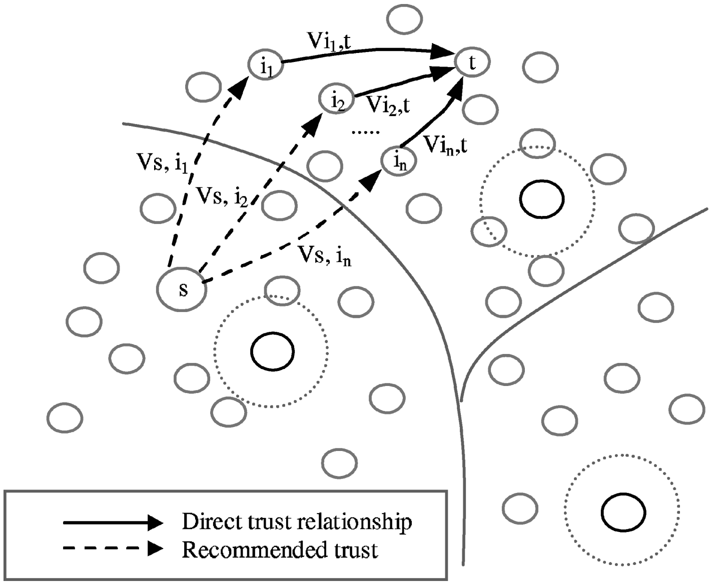


Fig. 9. Trust value update.

Ngai et al. [2004] simulated their scheme in GloMoSim [Zeng et al. 1998]. The 100 nodes in the $600m \times 600m$ network were divided into five groups with varied mobility between 0–10 m/s. The objective of the simulation was to test the public key management scheme in the presence of malicious nodes. A percentage, m, of the nodes were set to be adversaries and therefore always returned false public keys and trust values. The scheme was not implemented in parallel with a clustering algorithm but was divided into *fixed* groups. It was thus assumed that the network clustering structure was already built. If the scheme was to be implemented on top of the clustering algorithm (for example, the zonal algorithm [Chen and Liestman 2003]), it can be anticipated that the simulation would produce significantly different results. Schemes should always be simulated in a realistic setting, that is, performance can only be determined if the protocol under investigation is simulated together with the schemes central to its operation. These additional schemes in turn may consume additional network resources, which will certainly influence the results obtained from the simulations.

The cluster-based key management scheme does not alleviate the disadvantages of the certificate chaining approach [Capkun et al. 2003b] (see Section 9.3).

Key management schemes in MANETs should not rely on the functionality and correct operation of other schemes. A cluster-based key management approach relies on the effectiveness of a clustering scheme. Attacks on the clustering protocol thus introduce additional vulnerabilities and may create an indirect way of compromising the key management scheme and consequently the network's security mechanisms as a whole.

11. PREDEPLOYMENT-BASED KEY MANAGEMENT

The limited memory, energy, and computational power of sensor nodes result from the constraints placed on their cost and physical dimensions. In uncontrolled deployment, which is normally the case with large-scale sensor networks, the nodes are randomly scattered over the target area. This implies an unpredictable network topology

[Chan et al. 2003]. Furthermore, sensors can be added and subtracted after deployment and may also encounter hostile networking environments [Eschenauer and Gligor 2002]. These characteristics result in key predistribution being the only known, practical key establishment technique suitable for large-scale, wireless sensor networks [Eschenauer and Gligor 2002]. Sensor networks therefore fall under *authority-based* MANETs, as defined in Section 1.

11.1. System Model

Eschenauer and Gligor [2002] proposed a random key predistribution scheme for distributed sensor networks. An offline authority loads each sensor node with a large pool of keys prior to sensor deployment. Once in the operational environment, two nodes find a common key in their pools and use the shared key to secure subsequent communication. The scheme does not guarantee a shared key between all node pairs, but rather supports the existence of such a key with some probability. Nodes which cannot find a common key can establish a security association via a sequence of secure connections.

The key management scheme consists of three parts: key distribution, key revocation, and rekeying. The key distribution mechanism is further subdivided into three phases: key predistribution, shared-key discovery, and path-key establishment. In Eschenauer and Gligor [2002], specialized controller nodes provided key revocation services to the sensor network.

11.2. System Analysis

11.2.1. Key Distribution. The *key predistribution* phase consists of the following *offline* steps:

—The offline authority A generates a large pool P of keys and their identifiers.

—For each sensor, A draws k keys randomly from P without replacement. The k keys form the *key ring* of the sensor.

—A loads each sensor with its key ring and key identifiers.

—The key identifiers of each key ring and the associated sensor node's identity are stored on a trusted node.

—Finally, A loads the ith controller node with a shared key, where $K^{ci} = E_{K_x}(ci)$. $K_x = K_1 \oplus, \ldots, \oplus K_k$. K_i denotes the keys of a node's key ring and the controller's identity is given by ci. E_{K_x} is an encryption with node key K_x.

The *shared-key discovery* phase commences after deployment of the sensors. Each node broadcasts its key identifiers in plain text. Nodes within the transmission range discover shared keys by means of comparison. Alternatively, nodes can hide key-sharing patterns allowing for private shared-key discovery. The example given by Eschenauer and Gligor [2002] works as follows: each node broadcasts a list $\alpha, E_{K_i}(\alpha)$, where α is a random challenge and K_i for $i = 1, \ldots, k$ is all the keys in the node's key ring. Decryption of $E_{K_i}(\alpha)$ with the correct key will reveal α and warrants the establishment of a shared key with the broadcasting node.

The probability that two nodes share a common key on their key rings was given by Eschenauer and Gligor [2002]:

$$\frac{k!(P-k)!(P-k)!}{P!k!(P-2k)!}. \tag{3}$$

On completion of the shared-key discovery phase, nodes can set up a connected graph of secure links.

The *path-key establishment* phase is used to associate a path key between nodes within the transmission range which failed to establish a shared key during the *shared-key discovery* phase. If the graph is connected, a node can find a path to the neighboring node with which it failed to discover a common key [Chan et al. 2003]. The node can generate a path key and send the key to the neighboring node via one or more trusted intermediate nodes [Chan et al. 2003].

11.2.2. Revocation. The controller nodes (ci) have enhanced capabilities such as mobility and long transmission ranges. To revoke the keys of a specific node, a controller node broadcasts a signed list of the target node's key identifiers. A key K_e is used to sign the message. The controller unicasts an encryption of K_e to all nodes. K^{ci}, which is shared between the node and the controller, is used to encrypt K_e. The nodes decrypt $E_{K^{ci}}(K_e)$ and use K_e to verify the signature on the key identifier list. Nodes locate and remove the affected keys from their key rings by themselves. If the revocated keys affect any other secure links, a node restarts the *shared-key discovery* phase and if needed the *path-key establishment* phase.

11.2.3. Rekeying. Rekeying is equivalent to self-revocation; hence expired keys are removed from a node's key ring. In the case where one or more secure connections are lost, the node restarts the *shared-key discovery* phase and possibly the *path-key establishment* phase.

11.3. Discussion and Comments on Key Predistribution

The scheme proposed in Eschenauer and Gligor [2002] identifies key predistribution as the most practical technique to establish secure connectivity in sensor networks. After a node performs the *shared-key discovery* and *path-key establishment* phases, the probabilistic nature of the key distribution mechanism may leave a node with a partially connected graph. Chan et al. [2003] proposed that a node should increase its transmission range or request neighboring nodes to share their communications for a small number of hops. The mechanism, called *range extension*, requires nodes to gradually increase their transmission power and repeat the *shared-key discovery* and *path-key establishment* phases until they are fully connected. Chan et al. [2003] further improved on Eschenauer and Gligor [2002] and proposed a *q-composite* random key predistribution scheme which enhances network resilience against weaker adversaries. This, however, comes with greater vulnerability against large-scale attacks. In contrast to Eschenauer and Gligor [2002], nodes must find q common keys before they can establish secure connectivity.

Du et al. [2003, 2005] combined the scheme in Blom [1985] with the random key predistribution approach presented in Eschenauer and Gligor [2002] and Chan et al. [2003]. In contrast to the single key space (used in Blom's scheme), Du et al. [2003, 2005] used multiple key spaces. Blom's scheme guarantees a *complete* graph, that is, any pair of nodes can establish a common key. Due to the extension of the single key space to multiple key spaces, Du et al. [2003, 2005] sacrificed full connectivity for (1) better resilience against node captures and (2) reduced information storage. Du et al. [2003, 2005] also showed that multiple key spaces allow for improved resilience over previous schemes [Eschenauer and Gligor 2002; Chan et al. 2003], using the same amount of memory.

Liu and Ning [2003a] improved on the resilience and scalability of Eschenauer and Gligor [2002] and Chan et al. [2003]. Liu and Ning [2003a] built their scheme on random key predistribution and polynomial-based key predistribution [Bundo et al. 1993] and the scheme is essentially equivalent to that of Du et al. [2003]. The scheme allows any group of v parties to establish a common key. The key is perfectly secret with respect to a coalition of t other parties [Du et al. 2005]. Du et al. [2005] pointed out that Bundo et al. [1993] only achieved the lower bound on memory storage if all groups of size v can compute a common key and if the network is resilient against at most t captured nodes. Liu and Ning [2003a] also proposed a *grid-based* key predistribution scheme. A setup server constructs an $m \times m$ grid with a set of $2m$ polynomials, where $m = \lceil \sqrt[n]{N} \rceil$. The parameter n is the dimension of the hypercube and N the total number of sensors in the network. Each sensor is assigned a unique intersection on the grid and loaded with the polynomial shares of the intersecting polynomials. By using these shares, sensor nodes can perform polynomial share discovery and path-key discovery. Liu et al. [2005b], the journal version of Liu and Ning [2003a], extended the grid-based scheme to a multidimensional, *hypercube-based* scheme. An interesting feature of the scheme is that it can provide perfect resilience against node captures, for example, when $n = 4$ and $N = 20{,}000$.

It is clear that the *random subset assignment* key predistribution schemes of Liu and Ning [2003a], Liu et al. [2005b], and Du et al. [2003, 2005], compared to previous efforts, significantly enhance the resilience of key predistribution. However, as pointed out in Zhou et al. [2005], Liu et al. [2005a], and Chan and Perrig [2005], two main properties of these *random subset assignment* methods can be improved: (1) after a certain fraction of the sensors have been compromised, the fraction of compromised links between noncompromised nodes increases exponentially if any more nodes are compromised and (2) in order to ensure a connected graph these schemes require a sensor deployment with sufficient density. Sufficient node density cannot be guaranteed under the assumption that sensor nodes may fail (due to factors such as battery depletion) or sparse sensor deployment or as a result of node compromise [Zhou et al. 2005]. Although the grid-based scheme presented in Liu and Ning [2003a] and Liu et al. [2005b] may cope with poor connectivity, it also shows an exponential degradation in security after a certain fraction of nodes have been compromised.

PIKE [Chan and Perrig 2005] improves on the sensor density requirement of the *random subset assignment* key predistribution schemes by using an approach similar to the *grid-based* scheme proposed by Liu and Ning [2003a] and Liu et al. [2005b]. Each of the n nodes shares a unique pairwise key with $\sqrt{n}$ other sensors ($\lceil \sqrt{n} \rceil$ if n is not square). Nodes that do not share preloaded pairwise keys use trusted intermediate nodes to establish path keys [Chan and Perrig 2005]. Zhou et al. [2005] cited PIKE's network-wide communication to establish path keys as unsuitable for large-scale sensor networks.

Other schemes [Liu and Ning 2003b; Du et al. 2004; Huang et al. 2004] use *deployment information* to improve on the probabilistic key predistribution methods. All these approaches, however, assume that the deployment locations of the sensor can be predetermined to some extent [Zhou et al. 2005; Liu et al. 2005a].

Recent proposals [Zhou et al. 2005; Liu et al. 2005a] have argued that accurate a priori knowledge of sensor locations is unlikely in practice. Zhou et al. [2005] exploited a group-based deployment model to improve on the performance and resilience of the probabilistic approaches. As previous schemes suggest [Liu and Ning 2003b; Du et al. 2004; Huang et al. 2004], the probability that sensors in the same group are neighbors after deployment is high. The main idea is to preload each sensor with a carefully selected set of keys. Each sensor pair in the same group shares a common key. The key preloading technique ensures that after deployment groups are securely linked via associated nodes; each node can share a unique common key with up to t agents in any other group. Nodes that are not within the same group will thus use at most two trusted intermediaries to set up a path key. Liu et al. [2005a] proposed a group-based deployment model to eliminate the need for expected location information and to enhance the security and performance of the existing key predistribution schemes. The general framework requires nodes to be preloaded with an *in-group* key predistribution instance D_i. This allows nodes within the same group (which will most likely be neighbors after deployment) to use direct key establishment techniques. Nodes are also preloaded with a *cross-group* key predistribution instance D'_i. Nodes with the same D'_i form a *cross-group*. Nodes that are not within the same group need to find a "bridge" between their respective groups. The bridges are formed by two sensors from the same cross-group.

The existing literature on key management for sensor networks gives the impression of a thoroughly researched subject. In the view of the authors, the key predistribution field for sensor networks currently requires a comprehensive analysis of the existing schemes in terms of security, performance, and implementation practicality.

12. MOBILITY-BASED KEY MANAGEMENT APPROACHES

Capkun et al. [2003a, 2006] proposed mobility-assisted key establishment schemes for MANETs. As mentioned before, the authors of this survey view these schemes as a significant advance in the state of the art: in contrast to the previously discussed subsets, the protocols in Capkun et al. [2003a, 2006] introduce a shift in paradigm with respect to previous attempts to provide key management for MANETs. Most of the existing key management schemes for MANETs try to modify solutions suited for conventional wireline networks which may not always be ideal in MANETs. The proposals investigated Capkun et al. [2003a, 2006] are peer-to-peer key establishment schemes that rely on user mobility to bring nodes within each other's transmission range. This allows them to exchange their keying material without relying on a secure routing infrastructure. This effectively breaks the routing-security interdependence cycle [Bobba et al. 2003]. The remainder of the section focuses on the key agreement techniques proposed in Capkun et al. [2003a, 2006] for *fully* self-organized MANETs.

12.1. System Model

Capkun et al. [2003a, 2006] considered two models: a fully self-organized model and one with an offline trusted authority. The latter, an *authority-based* approach, is considered in the discussion (Section 12.3).

If a public key cryptosystem is used, two users u and v share a two-way security association if they exchange their triplets (U, k_u, a_u) and (V, k_v, a_v), where (U, V) are the names of users u and v, (k_u, k_v) are their public keys, and (a_u, a_v) are their respective node addresses. In a symmetric key setting, the public keys are replaced by a shared key k_{uv}.

Users are equipped with wireless nodes with an integrated side channel (such as an infrared interface). The side channel is used to set up security associations when users physically meet in the network. This inherently constitutes visual authentication by the users and allows users to bind user names to keying material. The security association setup mechanism can be enhanced through the use of *friend* nodes [Capkun et al. 2006].

To explain the key establishment mechanisms of Capkun et al. [2003a, 2006] in more detail, the establishment of security associations in both a public key and symmetric key setting is considered.

12.2. System Analysis

12.2.1. Public Key Approaches. Figure 10 illustrates the three main mechanisms for key establishment proposed in Capkun et al. [2003a, 2006]. The first [Mechanism (a)]

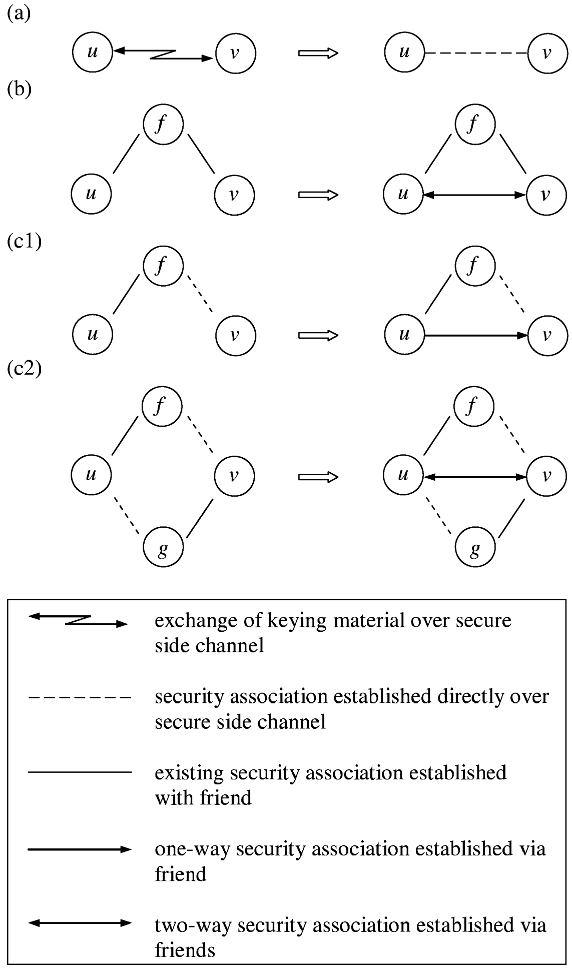

(a)
u
v
(b)
f
(c1)
(c2)
g
exchange of keying material over secure side channel
security association established directly over secure side channel
existing security association established with friend
one-way security association established via friend
two-way security association established via friends

Fig. 10. Direct and friend-assisted security association establishment.

allows users to establish a security association directly over the secure side channel during a physical encounter. The side channel ensures data integrity by eliminating the active adversary. Coupling Mechanism (a) with key confirmation and a defense against replay attacks results in Protocol 1 [Capkun et al. 2003a, 2006] detailed below:

Protocol 1 : Mechanism (a)

msg1 (secure side channel)	$u \rightarrow v$:	$a_u \parallel [\xi_u = h(r_u \parallel U \parallel K_u \parallel a_u)]$
msg2 (secure side channel)	$v \rightarrow u$:	$a_v \parallel [\xi_v = h(r_v \parallel V \parallel K_v \parallel a_v)]$
msg3 (radio channel)	$u \rightarrow v$:	$r_u \parallel U \parallel K_u \parallel a_u$
msg4 (radio channel)	$v \rightarrow u$:	$r_v \parallel V \parallel K_v \parallel a_v$
	u:	$h(r_v \parallel V \parallel K_v \parallel a_v) = \xi_v?; V?; match(K_v, a_v)$
	v:	$h(r_u \parallel U \parallel K_u \parallel a_u) = \xi_u?; U?; match(K_u, a_u)$
msg5 (radio channel)	$u \rightarrow v$:	$\sigma(r_v \parallel U \parallel V)$
msg6 (radio channel)	$v \rightarrow u$:	$\sigma(r_u \parallel V \parallel U)$

In msg 1 and msg 2, the users exchange their addresses (a_u, a_v) and the hash values (ξ_u, ξ_v) of their random numbers and triples. They need each other's address in order to exchange keying material on the radio interface in the following rounds. In msg 3 and 4, users exchange their triplets and random numbers over the radio interface. Each node checks whether the hash of the received random numbers and triplets over the radio link match the received hash values received over the side channel. In the final two messages, users send each other a signature $(\sigma())$ over the radio link. Verifying the signatures with the received public keys serve as proof that u and v knows the corresponding private keys. Note that the triplets and signatures could have been exchanged purely over the side channel, but Protocol 1 minimizes the amount of data sent over the side channel by sending hashes (ξ_u, ξ_v) as an integrity check code over the side channel. This allows u and v to exchange the rest of the information over the radio interface with the possibility of the man-in-the-middle attack eliminated.

Mechanism (b) uses a common friend f to generate and distribute to each node fresh certificates. Since f shares keying material with both u and v, the user can verify the received certificates from *f*.

Mechanism (c1) enhances key agreement during physical encounters with friend security associations and is simply a combination of Mechanism (a) and (b) detailed above. This mechanism can be used to establish either a one-way or two-way security association between u and v.

Mechanism (c2) will be discussed in the following section as it is more applicable in a symmetric key setting.

12.2.2. Symmetric Key Approaches. In a symmetric key setting, the three mechanisms illustrated in Figure 10 remain applicable in a different context.

With Mechanism (a), the users use the side channel to exchange all the necessary keying material to set up a shared key between them. The side channel in the symmetric key setting must also provide confidentiality in addition to data integrity. To avoid attack from passive adversaries, the users must be cautioned to activate their side channels with no other users within a "secure range" from them.

In Mechanism (b), users have a common friend f that plays the role of a trusted authority or trusted intermediary. There are well-established protocols that can be used in such a setup [Menezes et al. 1996].

Mechanism (c2) can be used if u and v do not share a common friend and in case they do not want the trusted third party to know their shared key. A friend of u, named *f*, and a friend of v, named g, are used as two separate paths by u and v to exchange key contributions. Protocol 2 [Capkun et al. 2003a, 2006] presented below explains Mechanism (c2) in more detail:

Protocol 2 : Mechanism (c2)

msg1	$u \rightarrow v$:	f, r_u
msg2	$v \rightarrow u$:	g, r_v
msg3	$u \rightarrow g$:	$u, \{d_{u\rightarrow g}, request, v, k_u, r_v\}k_{ug}$
msg4	$g \rightarrow v$:	$g, \{d_{g\rightarrow v}, reply, u, k_u, r_v\}k_{vg}$
msg3′	$v \rightarrow f$:	$v, \{d_{v\rightarrow f}, request, u, k_v, r_u\}k_{vf}$
msg4′	$f \rightarrow u$:	$f, \{d_{f\rightarrow u}, reply, v, k_v, r_u\}k_{uf}$
	u, v:	$k_{uv} = h(k_u \parallel k_v)$

In Protocol 2, users u and v use messages 1 and 2 to exchange random numbers (r_u, r_v) and the names (f, g) of their friends. In messages 3 and 4 and messages 3′ and 4′, u sends k_u to v via g and while v sends k_v to u via f. All the messages are encrypted with a shared symmetric key k_{xy}, where $x \in [u, v]$ and $y \in [g, f]$. In order to avoid ambiguity, each message includes the direction d and purpose of the message (*request* or *reply*). Users u and v generate a shared key k_{uv} by taking the hash of the concatenation of their individual contributions.

12.3. Discussion and Comments on Mobility-Based Key Management Approaches

The main characteristic of ad hoc networks is the lack of infrastructure. Nodes, therefore, are responsible for all network functionality of which routing is the most important. In stationary ad hoc networks, nodes may experience frequent link breakages as the traffic in the network increases. Node mobility significantly increases the frequency of these link breakages. This sporadic connectivity results in a poor availability feature and a high communication overhead for key management schemes that rely on the routing infrastructure. Another reason why relying on the routing infrastructure is infeasible is that any attack on the routing protocol may render the key management scheme insecure. The routing-security interdependence cycle [Bobba et al. 2003], in any case, forces the key management scheme to be independent of the routing mechanism. Clearly getting around this problem requires a complete shift with respect to the key management solutions found in conventional wireline networks. The fact that fully self-organized networks do not support any form of trusted authority, not even during offline initialization, adds a new dimension to this problem. Capkun et al. [2003a, 2006] detached the key management scheme from the routing infrastructure by exploiting user mobility. The mobility characteristic of MANETs, which are widely regarded as a limiting factor, are turned around as an aid to the key establishment mechanisms.

Dependence on mobility to bring users within a "secure range" in order for them to use their secure side channels for key establishment is the major disadvantage of Capkun et al. [2003a, 2006]. The authors themselves noted that it may take some time to set up a sufficient number of security associations. Their simulation results show that, as intuitively expected, the convergence time decreases with an increase in user mobility. The proposal will thus find a strong application as a complementary solution to other key management solutions and is ideally suited to establish security associations on the *application layer* in a self-organized setting [Capkun et al. 2006].

13. PARALLEL KEY MANAGEMENT APPROACHES

Yi and Kravets [2004] proposed a multiple key management approach by combining a distributed certificate authority (Section 9) and certificate chaining (Section 6). The proposal known as *composite* key management is based on two fundamental principles. First, key management should be shared between multiple nodes, and second, a trusted third party is required as an anchor of trust. Here certificates, as proposed in Capkun et al. [2003b], are stored and distributed by nodes in a self-organized nature. Yi and Kravets [2004] showed how a DCA can be used in *parallel* with certificate chaining to eliminate some of the weaknesses of the certificate chaining approach. The approach increases availability of the key management service since nodes can use either service to obtain keying material.

The remainder of the discussion on the proposals presented in Yi and Kravets [2004] will focus on the modifications and additional mechanisms added to certificate chaining and the DCA approach.

13.1. System Analysis

13.1.1. Metrics of Authentication. By introducing *authentication metrics* (confidence values), an attempt is made to provide users with a tool to calculate the level of trust that

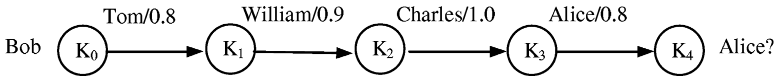


Fig. 11. Certificate chaining example.

they can place in an instance of authentication. Users thus assign a confidence value to certificates based on their relationship with the certificate owner. In Yi and Kravets [2004], the confidence value was also extended to incorporate the DCA as a trusted third party.

13.1.2. Trust Model. In the certificate chaining approach [Capkun et al. 2003b], users form a chain of trust by issuing certificates to other nodes in the network with which they have some relationship or have adequate reason to trust the binding between the node's identity and public key. Yi and Kravets [2004] illustrated this concept by means of an example: if Alice trusts that Bob is the holder of a public/private pair, Alice issues a certificate containing Bob's *ID*, Bob's public key, and other attributes such as a certificate lifetime parameter. Alice then generates a digital signature on Bob's certificate vouching for the certificate's authenticity.

Similarly to Capkun et al. [2003b], composite key management captures trust relationships between nodes in a certificate graph where the edges represent a digital certificate and vertices public keys. The edges are also coupled with a confidence value set by the certificate issuer and assigns a level of trust to the issued certificate.

An example of a certificate chain is given in Figure 11 [Yi and Kravets 2004]. In the example, if Bob wants to authenticate Alice, Bob needs to calculate a confidence value for the entire chain length. This is done by first multiplying all the confidence values for each edge together to form what is called a *raw confidence value*. To get the final confidence value, the chain length d and probability p of the nodes in the chain being compromised must also be considered. The raw confidence value thus needs to be multiplied by an attenuation factor $(1 - p)^{d-1}$ which yields the final confidence value for the chain as a whole. The user utilizes the final confidence value to make a decision on whether to grant the authentication or reject the chain as a possible authentication path.

13.1.3. Security Level of DCA. Where certificates in the certification graph are assigned an authentication metric, the DCA is assigned a security level (SL) reflecting the probability that an adversary can compromise the DCA. The security level is calculated as follows:

$$SL = 1.0 - \frac{\binom{n}{k}}{\binom{M}{c}}, \tag{4}$$

where n is the number of server nodes in the CA, k the *crypto threshold*, M the total number of nodes in the network, and c the number of nodes the most powerful adversary can compromise in a fixed time frame.

A system model example is given in Figure 12 [Yi and Kravets 2004] showing the composition of certificate chaining and a DCA. The certificates are assigned authentication metrics and the DCA a security level, as explained above.

13.2. Discussion and Comments on Parallel Key Management Approaches

The approach of merely combining the distributed certificate authority scheme [Zhou and Haas 1999; Yi and Kravets 2003] and certificate chaining scheme [Capkun et al. 2003b] fails to adequately address the key management problem in MANETs since the

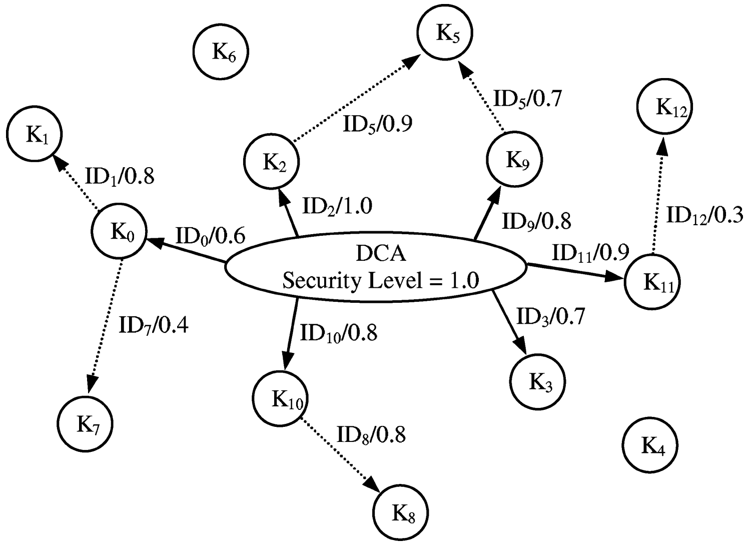


Fig. 12. Example system model showing DCA composed with one-hop certificate chaining.

composite key management scheme [Yi and Kravets 2004] inherits all the weaknesses of the distributed certificate authority approach as stipulated in Section 6.3. In fact, it further degrades the security as it only solves the availability problem of Capkun et al. [2003b] while inheriting its weak authentication property (see Section 9.3).

The scheme proposed in Capkun et al. [2003b] was designed for "open" or fully self-organized MANETs. Yi and Kravets [2004] claimed that they improved on Capkun et al. [2003b], but in fact Yi and Kravets [2004] were proposing a scheme for an entirely different application.

Combining two or more key management approaches to eliminate the disadvantages of the other will in most cases not be effective in MANETs: the authors have investigated a combination of *all* key management schemes referenced in this article. The mobility-based approaches (Section 12 [Capkun et al. 2003a, 2006]) may be added as a complement to some schemes to enhance the certificate exchange process by exploiting user mobility. For example, combining the mobility-based approach and the certificate chaining approach will clearly be more suitable than combining any one of the two with the distributed certificate authority approach: the disadvantages of the distributed certificate authority approach is, in the view of the authors, unavoidable (see Section 6.3).

Researchers should not be discouraged from looking at ways to combine the existing key management schemes, but take caution not to create new disadvantages in the process. A complex interaction between the two schemes may also close the window for a strong security argument.

14. CONCLUSIONS AND FUTURE DIRECTION

This article presented a survey on peer-to-peer or pairwise key management for mobile ad hoc networks (MANETs). Investigations by the authors within the published proposals have shown that the existing protocols can be grouped into several categories. Each category was discussed by introducing at least the original protocol from within the grouping. This way of categorizing the available protocols gives one the opportunity to establish a deeper insight into the available key management approaches in MANETs and to discuss the strengths and weaknesses of each.

The introduction to MANETs provided the necessary background required to follow the discussions and comprehend the comments on the key management protocols.

Conclusions have been separately provided for key management schemes designed for *fully self-organized* MANETs and schemes suitable for *authority-based* MANETs (see Section 1):

—From the reviewed key management protocols, the mobility-based approaches [Capkun et al. 2006] are the most feasible for *fully self-organized key management* on the application layer. The remaining obstacles to be eliminated are the dependence on mobility during the bootstrapping of the routing security and minimizing user interaction on the application layer. Approaches presented in Cagalj et al. [2006] and McCune et al. [2005] solve the latter problem by effectively reducing user operation down to the "push of a button"; users will not use security mechanisms that inconvenience them in anyway. In the view of the authors it is important to decouple key management mechanisms intended for securing application level services from those used to secure the routing infrastructure. A failure in the users' ability to judge the honesty or intent of other users should not jeopardize the security of any basic network service.

—Most of the *authority-based* approaches make use of an *online* authority in addition to the *offline* authority to provide important key management functions such as certificate renewal. From the discussions in Section 6.3 and Section 7.3, it is clear that the use of an online authority is problematic in MANETs, both in a partially or fully distributed form. The only solution that emerges is to completely eliminate any form of *online* authority. Capkun et al. [2006] proposed an *authority-based* scheme where each node is preloaded by the offline authority with a certificate. After network formation, each node becomes its own authority domain and distributes its certificate to nodes within its transmission range. This solution is unfortunately not complete; Capkun et al. [2006] did not address certificate renewal and revocation. Furthermore, the scheme is dependent on mobility and fails in a low mobility or stationary ad hoc network.

Key management schemes based on the key predistribution techniques proposed for sensor networks may be another avenue to solve the key management problem in authority-based MANETs.

Another observation is related to the criteria used by researchers to analyze key management schemes for MANETs. Key management schemes are designed either for an "open" (self-organized) or "closed" (authority-based) network and consequently aimed at different applications. "Open" or *fully* self-organized MANETs have some inherent security implications (such as being vulnerable against the Sybil attack [Douceur 2002]) and must be analyzed accordingly. It is therefore not always possible to compare schemes that assume the existence of a trusted authority with those that are fully self-organized.

This study confirms that key management mechanisms proposed to guarantee the security of conventional networks are not necessarily suitable or adaptable to MANETs. Novel techniques, designed specifically for MANETs, are necessary.

Key management is an important area that will need resolution before wide-scale deployment of ad hoc networks will become practical. Although key management for MANETs has reached a reasonable level of maturity, it is still a research area with room for innovation.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

Abdul-Rahman, A. and Hailes, S. 1997. A distributed tust model. In *Proceedings of the ACM New Security Paradigms Workshop*.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci. 2002. A survey on sensor networks. *IEEE Commun. Mag. 40*, 8 (Aug.), 102–114.

Ateniese, G., Steiner, M., and Tsudik, G. 1998. Authenticated group key agreement and friends. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*.

Ayanoglu, E., I, C.-L., Gitlin, R. D., and Mazo, J. E. 1993. Diversity coding for transparent self-healing and fault-tolerant communication networks. *IEEE Trans. Commun. 41*, 11, 1677–1686.

Beth, T., Malte, B., and Birgit, K. 1994. Valuation of trust in open networks. In *Proceedings of the Third European Symposium on Research in Computer Security*.

Blom, R. 1985. An optimal class of symmetric key generation systems. In *Proceedings of EUROCRYPT'84*.

Bobba, R. B., Eschenauer, L., Gligor, V. D., and Arbaugh, W. 2003. Bootstrapping security associations for routing in mobile ad-hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference*.

Boneh, D. and Franklin, M. 2001. Identity-based encryption from weil pairing. In *Proceedings of the Conference on Advances in Cryptology* (CRYPTO'01).

Broch, J. and Johnson, D. B. 1999. The dynamic source routing protocol for mobile ad hoc networks. IETF Internet Draft. October.

Bundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., and Yung, M. 1993. Perfectly-secure key distribution for dynamic conferences. In *Proceedings of CRYPTO'92*.

Buttyan, L. 2001. Building blocks for secure services: Authenticated key transport and rational exchange protocols. Ph.D. dissertation. Universite Technique de Budapest, Budapest, Hungary.

Buttyan, L. and Hubaux, J. P. 2003. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM Mobile Netw. Appl. 8*, 5, 579–592.

Cagalj, M., Capkun, S., and Hubaux, J. 2006. Key agreement in peer-to-peer wireless networks. *Proc. IEEE* (Special Issue on Cryptography and Security) *94*, 2, 467–478.

Capkun, S., Buttyan, L., and Hubaux, J.-P. 2003a. Mobility helps security in ad hoc networks. In *Proceedings of MobiHoc*.

Capkun, S., Buttyan, L., and Hubaux, J.-P. 2003b. Self-organized public-key management for mobile ad hoc networks. *IEEE Trans. Mobile Comput. 2*, 1, 52–64.

Capkun, S., Hubaux, J., and Buttyan, L. 2006. Mobility helps peer-to-peer security. *IEEE Trans. Mobile Comput. 5*, 1, 43–51.

Carter, C., Yi, S., Ratanchandani, P., and Kravets, R. 2003. Manycast: Exploring the space between anycast and multicast in ad hoc networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking* (MOBICOM'03).

Cha, J. C. and Cheon, J. H. 2003. An identity-based signature from gap diffie-hellman groups. In *Proceedings of the Conference on Public Key Cryptography* (PKI'03).

Chan, A. C.-F. 2004. Distributed symmetric key management for mobile ad hoc networks. In *Proceedings of the 23rd Conference of the IEEE Communications Society*.

Chan, H. and Perrig, A. 2005. PIKE: Peer intermediaries for key establishment in sensor networks. In *Proceedings of INFOCOM'05*.

Chan, H., Perrig, A., and Song, D. 2003. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Symposium of Privacy and Security*.

Chen, Y. P. and Liestman, A. L. 2003. A zonal algorithm for clustering ad hoc networks. *Int. J. Foundat. Comput. Sci. 14*, 2, 305–322.

Christianson, B. 1996. Why isn't trust transitive. In *Proceedings of the International Workshop on Security Protocols*.

Dahill, B., Levine, E., Royer, E., and Shields, C. 2001. A secure routing protocol for ad hoc networks. Tech. rep. UM-CS-2001-037. University of Massachusetts, Amherst, MA.

Dearham, N. J. 2003. Development, implementation and quantification of an ad-hoc routing protocol for mobile handheld terminals. M. S. thesis in Electronic Engineering. Department of Electrical, Electronic and Computer Engineering, University of Natal, Durban, South Africa.

Deng, H., Mukherjee, A., and Agrawal, D. P. 2004. Threshold and identity-based key management and authentication for wireless ad hoc networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing* (ITCC'04).

Desmedt, Y. and Jajodia, S. 1997. Redistributing secret shares to new access structures and its applications. Tech. rep. ISSE-TR-97-01. Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Fairfax, VA.

Dolev, D. and Yao, A. C. 1983. On the security of public key protocols. *IEEE Trans. Inform. Theor. 29*, 2, 198–208.

Douceur, J. R. 2002. The Sybil attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems* (IPTPS'02).

Du, W., Deng, J., Han, Y., Chen, S., and Varshney, P. 2004. A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of INFOCOM'04*.

Du, W., Deng, J., Han, Y. S., and Varshney, P. K. 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security* (CCS).

Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., and Khalili, A. 2005. A pairwise key pre-distribution scheme for wireless sensor networks. *ACM Trans. Inform. Syst. Secur. 8*, 2, 228–258.

Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communication Security* (CCS'02).

FEM 2005. U.S Federal Emergency Management Agency (FEMA): Information on federally declared disasters. Available online at `http://www.fema.gov`.

Fouque, P.-A. and Stern, J. 2001. One round threshold discrete-log key generation without private channels. In *Proceedings of the Public Key Cryptography* (PKC'01).

Franz, W., Eberhardt, R., and Luckenbach, T. 2001. Fleenet—Internet on the road. In *Proceedings of the 8th World Congress on Intelligent Transport Systems*.

Gennaro, R., Jarecki, S., Krawczyk, H., and Rabin, T. 1999. Secure distributed key generation for discrete-log based cryptosystems. In *Proceedings of the Conference on Advances in Cryptology* (EUROCRYPT'99).

Haas, Z. J., Deng, J., Liang, B., Papadimitratos, P., and Sajama, S. 2002. Wireless ad hoc networks. In *Encyclopedia of Telecommunications*, J. Proakis, Ed. John Wiley, New York, NY.

Haas, Z. J. and Perlman, M. 1998. The performance of query control schemes for zone routing protocol. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication* (SIGCOMM'98).

Haas, Z. J. and Tabrizi, S. 1998. On some challenges and design choices in ad-hoc communications. In *Proceedings of the IEEE Military Communications Conference* (MILCOM'98).

Herzberg, A., Jaracki, S., Krawczyk, H., and Yung, M. 1995. Proactive secret sharing or: How to cope with perpetual leakage. In *Proceedings of the Conference on Advances in Cryptology* (CRYPTO'95).

Hu, Y.-C., Johnson, D. B., and Perrig, A. 2002a. Ariadne: A secure ondemand routing protocol for ad hoc networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking* (Mobicom'02).

Hu, Y.-C., Johnson, D. B., and Perrig, A. 2002b. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications.*

Huang, D., Mehta, M., Medhi, D., and Harn, L. 2004. Location-aware key management scheme for wireless sensor networks. In *Proceedings of the ACM Workshop on Security for Ad Hoc and Sensor Networks* (SASN).

Hubaux, J.-P., Buttyan, L., and Capkun, S. 2001. The quest for security in mobile ad hoc networks. In *Proceedings of MobiHoc'01*.

Johnson, D. B. and Maltz, D. A. 1996. Dynamic source routing in ad-hoc wireless networks. In *Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 153–181.

Josang, A., Gray, E., and Kinateder, M. 2003. Analysing topologies of transitive trust. In *Proceedings of the First International Workshop on Formal Aspects in Security and Trust* (FAST'03).

Joshi, D., Namuduri, K., and Pendse, R. 2005. Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: An analysis. *EURASIP J. Wireless Commun. Netw.* 4, 579–589.

Joye, M. and Yen, S.-M. 1998. ID-based secret-key cryptography. *ACM Operat. Syst. Rev. 32*, 4, 33–39.

Jubin, J. and Tornow, J. D. 1987. The DARPA Packet Radio Network Protocol. *IEEE 75*, 1, 21–32.

Khalili, A., Katz, J., and Arbaugh, W. A. 2003. Towards secure key distribution in truly ad-hoc networks. In *Proceedings of the IEEE Workshop on Security and Assurance in Ad-Hoc Networks*.

KIM, Y., PERRIG, A., AND TSUDIK, G. 2000. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the 7th ACM Conference on Computer and Communications Security* (CCS'00).

KIM, Y., PERRIG, A., AND TSUDIK, G. 2004. Tree-based group key agreement. *ACM Trans. Inform. Syst. Sec. 7*, 1, 60–96.

KONG, J., ZERFOS, P., LUO, H., LU, S., AND ZHANG, L. 2001. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Proceedings of the Ninth International Conference on Network Protocols* (ICNP'01).

LEE, W.-B. AND CHANG, C.-C. 1999. (t, n) Threshold digital signature with traceability property. *J. Inform. Sci. Eng. 15*, 5, 669–678.

LI, C.-M., HWANG, T., AND LEE, N.-Y. 1994. Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders. In *Proceedings of the Conference on Advances in Cryptology* (EUROCRYPT'94).

LI, Z.-C., ZHANG, J.-M., LUO, J., SONG, W., AND DAI, Y.-Q. 2001. Group-oriented (t, n) threshold digital signature schemes with traceable signers. In *Proceedings of Topics in Electronic Commerce, Second International Symposium* (ISEC 2001).

LIU, D. AND NING, P. 2003a. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*.

LIU, D. AND NING, P. 2003b. Location-based pairwise key establishments for static sensor networks. In *Proceedings of the ACM Workshop on Security for Ad Hoc and Sensor Networks* (SASN).

LIU, D., NING, P., AND DU, W. 2005a. Group-based key pre-distribution in wireless sensor networks. In *Proceedings of the ACM Workshop on Wireless Security* (WiSe'05).

LIU, D., NING, P., AND RONGFANG, L. 2005b. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inform. Syst. Sec. 8*, 1, 41–77.

LUO, H., ZERFOS, P., KONG, J., LU, S., AND ZHANG, L. 2002. Self-securing ad hoc wireless networks. In *Proceedings of the Seventh International Symposium on Computers and Communications* (ISCC'02).

MCCUNE, J. M., PERRIG, A., AND REITER, M. K. 2005. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proceedings of the IEEE Symposium on Security and Privacy*.

MENEZES, A., VAN OORSCHOT, P., AND VANSTONE, S. 1996. *Handbook in Applied Cryptography*. CRC Press, Boca Raton, FL.

MICHELS, M. AND HORSTER, P. 1996. On the risk of disruption in several multiparty signature schemes. In *Proceedings of the Advances in Cryptology* (ASIACRYPT'96).

MORRIS, R., JANNOTTI, J., KAASHOEK, F., LI, J., AND DECOUTO, D. 2000. Carnet: A scalable ad hoc wireless network system. In *Proceedings of the 9th ACM SIGOPS European Workshop*.

NGAI, E. C. H., LYU, M. R., AND CHIN, R. T. 2004. An authentication service against dishonest users in mobile ad hoc networks. In *Proceedings of the IEEE Aerospace Conference*.

PAPADIMITRATOS, P. AND HAAS, Z. J. 2002. Secure routing for mobile ad hoc networks. In *Proceedings of the SCS Communication Network and Distributed System Modeling and Simulation Conference* (CNDS'02).

PARK, V. D. AND CORSON, M. S. 1997. A highly adaptable distributed routing algorithm for mobile wireless networks. In *Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM'97).

PEDERSEN, H. 1997. How to convert any digital signature scheme into a group signature scheme. In *Proceedings of the 5th International Workshop on Security Protocols*.

PERKINS, C. E. AND BELDING-ROYER, E. M. 1999. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA'99).

PETERSEN, H. AND HORSTER, P. 1997. Self-certified keys—concepts and application. In *Proceedings of the Third Conference on Communication and Multimedia Security*.

QUAZI, T. A.-M. 2003. Design and implementation of an on-demand ad-hoc routing algorithm for a positional communication system. M. S. thesis in Electronic Engineering, Department of Electrical, Electronic and Computer Engineering, University of Natal, Durban, South Africa.

RAVI, S., RAGHUNATHAN, A., KOCHER, P., AND HATTANGADY, S. 2004. Security in embedded systems: Design challenges. *ACM Trans. Embedd. Comput. Syst. 3*, 3, 461–491.

RAYA, M. AND HUBAUX, J. P. 2005. The security of vehicular ad hoc networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks* (SASN'05).

SALEM, N. B., BUTTYAN, L., HUBAUX, J.-P., AND JAKOBSSON, M. 2005. Node cooperation in hybrid ad hoc networks. *IEEE Trans. Mob. Comput. 5*, 4, 365–376.

SHAMIR, A. 1979. How to share a secret. *Commun. ACM 22*, 11, 612–613.

STEINER, M., TSUDIK, G., AND WAIDNER, M. 2000. Key agreement in dynamic peer groups. *IEEE Trans. Parall. Distrib. Syst. 11*, 8, 769–780.

STERBENZ, J. P. G., KRISHNAN, R., HAIN, R. R., JACKSON, A. W., LEVIN, D., RAMANATHAN, R., AND ZAO, J. 2002. Survivable mobile wireless networks: Issues, challenges, and research directions. In *Proceedings of the ACM Workshop on Wireless Security* (WiSe'02).

TAUB, H. AND SCHILLING, D. L. 1991. *Principles of Communication Systems*, 2nd Ed. McGraw-Hill, New Delhi, India.

TOH, C.-K. 2001. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, Englewood Cliffs, NJ.

TSENG, Y.-M. AND JAN, J.-K. 1999. Attacks on threshold signature schemes with traceable signers. *Inform. Process. Lett. 71*, 1, 1–4.

WANG, C.-T., LIN, C.-H., AND CHANG, C.-C. 1998. Threshold signature schemes with traceable signers in group communications. *Comput. Commun. 21*, 8, 771–776.

WANG, G., HAN, X., AND ZHU, B. 2003. On the security of two threshold signature schemes with traceable signers. In *Proceedings of Applied Cryptography and Network Security, First International Conference* (ACNS 2003).

WONG, T. M., WANG, C., AND WING, J. M. 2002. Verifiable secret redistribution for archive system. In *Proceedings of the First International IEEE Security in Storage Workshop*.

WU, B., WU, J., FERNANDEZ, E. B., ILYAS, M., AND MAGLIVERAS, S. 2005a. Secure and efficient key management in mobile ad hoc networks. *J. Netw. Comput. Appl.*

WU, B., WU, J., FERNANDEZ, E. B., AND MAGLIVERAS, S. 2005b. Secure and efficient key management in mobile ad hoc networks. In *Proceedings of the First International Workshop on Systems and Network Security* (SNS2005) (in conjunction with IPDPS).

WU, T.-S. AND HSU, C.-L. 2004. Cryptanalysis of group-oriented (t, n) threshold digital signature schemes with traceable signers. *Comput. Stand. Interfac. 26*, 5, 477–481.

XU, G. AND IFTODE, L. 2004. Locality driven key management architecture for mobile ad-hoc networks. In *Proceedings of the First IEEE International Conference on Mobile and Sensor Networks* (MASS'04).

YI, S. AND KRAVETS, R. 2001. Practical PKI for ad hoc wireless networks. Tech. rep. UIUCDCS-R-2002-2273, UILU-ENG-2002-1717. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL.

YI, S. AND KRAVETS, R. 2002a. Key management for heterogeneous ad hoc wireless networks. Tech. rep. UIUCDCS-R-2002-2290, UILU-ENG-2002-1734. Department of Computer Science, University of Illinois. at Urbana-Champaign, Urbana, IL.

YI, S. AND KRAVETS, R. 2002b. Key management for heterogeneous ad hoc wireless networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols* (ICNP'02).

YI, S. AND KRAVETS, R. 2003. MOCA: Mobile certificate authority for wireless ad hoc networks. In *Proceedings of the 2nd Annual PKI Research Workshop* (PKI 2003).

YI, S. AND KRAVETS, R. 2004. Composite key management for ad hoc networks. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (MobiQuitous'04).

ZENG, X., BAGRODIA, R., AND GERLA, M. 1998. GloMoSim: A library for parallel simulation for large-scale wireless networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulations* (PADS '98).

ZHANG, R. AND IMAI, H. 2003. Round optimal distributed key generation of threshold cryptosystem based on discrete logarithm problem. In *Proceedings of the Conference on Applied Cryptography and Network Security* (ACNS'03).

ZHOU, L. AND HAAS, Z. J. 1999. Securing ad hoc networks. *IEEE Netw*. (Special Issue on Network Security) *13*, 6, 24–30.

ZHOU, L., NI, J., AND RAVISHANKAR, C. V. 2005. Efficient key establishment for group-based wireless sensor deployments. In *Proceedings of the ACM Workshop on Wireless Security* (WiSe'05).

ZIMMERMANN, P. 1995. *The Official PGP User's Guide*. MIT Press, Cambridge, MA.

Received October 2004; revised February 2006, July 2006; accepted August 2006