Models of Authentications in Ad Hoc Networks and Their Related Network Properties

Katrin Hoeper and Guang Gong
{khoeper, ggong}@calliope.uwaterloo.ca
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

Abstract

There is still no consistent definition of general ad hoc network properties. All introduced protocols for ad hoc networks are based on different assumptions and security requirements, and are consequently suited for specific applications only. Due to the broad field of applications, a general security model can still not be found in any literature. We define two ad hoc network categories, namely mobile ad hoc networks (MANETs) and smart sensor network, and discuss all further definitions and observations separately for both implementations. The main contribution of this report is the clear definition of properties, parameters, architectures, security requirements, and authentication models of ad hoc networks. Furthermore, we derive design goals for all protocols to be implemented in ad hoc networks. We also provide an extensive overview of present and future ad hoc network applications, used standards, and proposed solutions.

We believe that the authentication of network nodes and the establishment of secret keys among nodes are both target security objectives in ad hoc networks. The constrained devices and other special properties of ad hoc networks make achieving those security properties a challenging task. We comprehensively discuss proposed protocols in each introduced authentication model, state their features and drawbacks, and identify their limitation of use. We show that providing entity authentication and authentic key exchange in ad hoc networks is a security problem still not satisfyingly solved. The next contribution of this paper is the introduction of a general protocol framework for the implementation of authentication and key establishment protocols in ad hoc networks. The combination of the derived design goals and the protocol framework enables us to examine existing ad hoc network protocols and to develop new authentication and key establishment protocols that are applicable in ad hoc networks while meeting the network's security requirements at the same time.

Keywords: Ad hoc network, sensor network, MANET, authentication, key establishment

1 Introduction

Recently many people in the media, industry, and academia are talking about ubiquitous computing and ad hoc networking, but it seems that everybody has a different understanding of the topic. Some people associate ad hoc networks with Personal Area Networks (PANs), as for instance wireless communications between PDA's, cellular phones, and laptops using the Bluetooth [7] protocol, whereas others might imagine military applications, such as exploring enemy territory by

Figure 1: One-to-one communication, e.g., beaming business cards from one PDA to another PDA.

Figure 2: Many-to-one communication, e.g. a PAN with many consumer devices talking to a laptop.

Figure 3: One-to-many communication, e.g. one remote control controls all home appliances.

the use of sensor networks. So what are ad hoc networks? What is their infrastructure? What are their properties? What are the applications of such networks and do those applications require the implementation of any security? All these questions have not been sufficiently answered yet. Clear definitions of architecture, properties, and security requirements can still not be found in any literature. Although some applications are already implemented, the desired security properties have still not been completely achieved.

1.1 Applications

We summarize some current and future applications of ad hoc networks in Table 1. The applications listed in the table are sorted by their area of use, network devices, and communication model. This table demonstrates the diversity of ad hoc networks, e.g. founded in their different architecture. The communication models of ad hoc networks include one-to-one, many-to-one, one-to-many, and any-to-any communications, as demonstrated in some sample applications in the Figures 1- 5. Some ad hoc networks are connected to a fixed backbone via access points, as shown in Figure 6, which is denoted as *fixed backbone* in the table. We cite some of the papers that introduce solutions for the respective application in the right-most column. The table makes clear it that different applications have different security requirements. As a consequence, an existing solution that works for one application is not necessarily suited for an implementation in another application.

1.2 Security Goals

We believe that there are four main security problems that need to be dealt with in ad hoc networks: (1) the authentication of devices that wish to talk to each other; (2) the secure establishment of a session key among the authenticated devices; (3) the secure routing in multi-hop networks; and (4) the secure storage of key data in the devices. The complexity and diversity of ad hoc networks has led to a variety of proposals which concentrate on different security problems. Most current research about security in ad hoc networks deals with secure routing. Solutions for efficient routing that deal with the dynamic network topology are introduced, for instance, in [43, 30]. The problem of selfish network nodes[1] is also addressed and solutions are introduced [11, 48]. To securely store key data, the network devices need to provide tamper resistant memory. There is also a lot of ongoing research on this topic [33, 1]. There are not many papers dealing with authentication and

[1] These are network nodes that would rather save their own battery power than forward somebody else's packets.

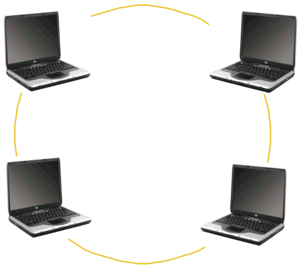

Figure 4: Any-to-any communication, e.g., laptops communicating at a conference.

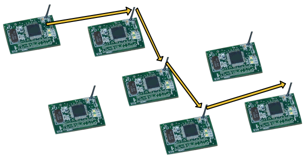

Figure 5: Any-to-any communication, e.g. any sensor can broadcast message and any sensor can forward it to base station.

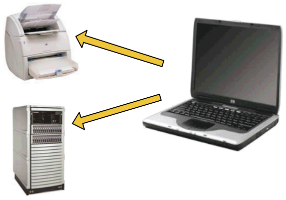

Figure 6: One-to-many communication, e.g. laptop has access to network resources (fixed backbone) via access point.

key exchange in ad hoc networks even though this problem has still not been satisfactorily solved for all potential applications.

Most published ad hoc network protocols assume that authentication of the devices has already taken place before the protocol execution. Consequently, most solutions assume that the devices share a secret a priori. Those papers never explain how authentication and key establishment are achieved even though the solutions are based on a successful authentication of the talking devices.

The solutions that are introduced for authentication and key establishment cover only special applications and cannot be generalized. We will focus on entity authentication and key establishment throughout this paper. There are many problems due to the nature of ad hoc networks, such as the dynamic infrastructure and the constrained network devices that make authentication a challenging task to achieve in such networks. For instance, we need to take into account network nodes might frequently join or leave the network. We will discuss all problems in greater detail later in this report.

1.3 Objectives and Outline

The first objective of this report is to define the different authentication models that exists for ad hoc networks and summarize previous work on authentication and key establishment protocols for each model. We review the introduced solutions and classify their suitability for an implementation in ad hoc networks.

The next objective of this report is the definition of a general architecture, properties, and parameters of ad hoc networks. These definitions help us derive design goals for ad hoc network protocols. Another contribution is the introduction of a framework for authentication and key establishment protocols in ad hoc networks. Having such a framework is necessary to develop protocols that are of a more general kind and not customized for a single application.

The remainder of this report is organized as follows. In the next section we introduce several authentication models and discuss proposed protocols in each model. In Section 3 we define the properties of ad hoc networks and their devices. In Section 4 we introduce some parameters of ad hoc networks. Later in the same section we derive design goals for authentication and key establishment protocols in ad hoc networks. In the last section we summarize our results, describe trends in ad hoc networks, and state the directions for our future investigations.

Table 1: Applications of ad hoc networks

Area of use	Who's talking	Communication model	Possible Applications
Civil applications	PDA to Laptop	one-to-one	Synchronize data, e.g., update calendar, etc. [7]
	PDA to PDA	one-to-one or one-to-many	Exchanging business cards [7]
	Laptop to Laptop	any-to-any	Network games [4], conference [47]
	Laptop to Shared Resource [fixed backbone]	one-to-one	Access to file server, printer, etc. from everywhere [7]
	Laptop to Laptop [fixed backbone]	one-to-many or any-to-any	Virtual classroom [59]
	Cell Phones to Base Station [fixed backbone]	one-to-many	Cellular phone system
	Laptop to Internet Access Point [fixed backbone]	one-to-many	Internet access at airports [4], trade fairs, (internet) cafes, museum items, bookstore items [27]
	Several Home Appliances to Wireless Access Point in Household [fixed backbone]	one-to-many	One remote control for HIFI, TV, VCR, electronic blinds, garage door, refrigerator, etc. [51]
Military services	Sensor to Sensor/Base Station	one-to-one or one-to-many	Rescue missions [59], data collection, smartDust [52], selfhealing minefields [28]
	Weapon to Owner	one-to-one	Restrict use of weapons [51]
Governmental applications	Sensor to Sensor	one-to-many or many-to-many	Law enforcement [59], Emergency scenarios [56]
Health services	Medical Device to Base Station [fixed backbone]	one-to-many	Thermometer, heart monitors, blood oxygen meters connected to doctor's palmtop or nursing station [51]

2 Authentication Models and Previous Work

In this section we discuss general authentication models and existing protocols for providing entity authentication and/or secure key establishment in ad hoc networks. We briefly describe the different models and discuss some proposed protocols for each of them. We point out advantages, drawbacks, and suited applications for each protocol. We summarize all models for better comparison in Table 2 in which the models are sorted by their used encryption scheme. We reference some papers that introduced protocols in the respective model in the right-most column.

2.1 No Authentication

Certain network protocols are introduced for use in ad hoc networks that do not support entity authentication and/or other security feature at all. For instance, the Piconet project [6] was implemented to study embedded mobile networking with focus on the connectivity among the network devices while omitting the implementation of any security. Other systems are mainly designed for a use in PANs, such as HomeRF [23] and IrDA [26]. In PANs one user wants to connect several of his/her personal devices. The HomeRF Working Group proposed the HomeRF Shared Wireless Access Protocol (SWAP) as an open industry specification to enable secure communication among consumer devices. Since in most scenarios we have only one user, authentication among the devices is not required and thus is not implemented in the protocol. The Blowfish algorithm with a 56 bit key size is used for confidential communication. It is obvious that such a key length is prone to brute-force attacks. The next generation of HomeRF will use 128 bit encryption. All devices share the same key which is set by default or by the user. Please note that the HomeRF Working Group was disbanded in January 2003 and their homepage is no longer available. The Infrared Data Association introduced the IrDA protocol which utilizes infrared instead of radio frequencies. The advantage of infrared over radio is the higher communication bandwidth, whereas the drawbacks are the required line of sight and the limited distance between the communicating devices. IrDA is recommended for PANs to connect wireless devices with the home computer. The implementation of any security is not considered in the protocol, but the very limited distance and the required line of sight among devices could be used to provide authentication.

2.2 Symmetric Solutions

When using symmetric encryption a secret must be shared among all devices that wish to communicate. The secret sharing can be achieved by transmitting a secret over a confidential and authentic channel prior to the execution of the authentication protocol. If we want to use the common secret directly to encrypt the communication, the parties that wish to communicate need to share a symmetric key of appropriate size, e.g. 128 bit. It is not recommended to use the same encryption key for a long period of time. To avoid this, a fresh session key can be derived from a common information and/or previous session keys in a pre-defined fashion. The secret key can be used for the authentication of the devices, for instance by executing a challenge and response protocol [40]. We know of two current standards and one ad hoc network protocol which are based on a pure symmetric scheme. There also exists a model for probabilistic key pre-distribution in sensor networks.

I-A. IEEE 802.11 (b) Model

The IEEE standard for wireless local area networks (WLAN) IEEE 802.11 [24] was approved in 1997. The standard is proposed to define the over-the-air interface between wireless clients and a base station, or between two wireless clients. Usually this standard is not considered for an implementation in ad hoc networks, the protocol is rather designed for networks with a fixed infrastructure. There are two authentication modes in the protocol: (1) *open system*, which is the default setting and does not require any form of authentication among the communicating devices; (2) *shared key*, which requires that the communicating parties exchange a secret key over a secure side-channel prior to the execution of the authentication protocol. Hence, the first mode does not provide any authentication and the second mode uses a simple challenge and response protocol to verify if both parties are in possession of the same key. The WEP algorithm is used in the authentication protocol as well as encryption algorithm. The algorithm requires an external key management, i.e. users need a secure side-channel to exchange the encryption keys. This requirement is the crucial point of the protocol because it can be very restrictive in some ad hoc networks. We cannot assume the existence of a secure side-channel among all devices in all ad hoc network applications. Furthermore, the implemented authentication protocol is weak and attacks are already presented, e.g. in [2]. The key size in the standard is 40 bits which is prone to brute-force attacks. The WEP algorithm is proven to be weak even when the key size would be increased and many attacks are introduced, e.g. [54, 10]. Just recently, on June 24, 2004, the IEEE ratified the 802.11 i security standard for wireless LAN. In this new standard the weak WEP algorithm is replaced by the AES. The improved authentication algorithm is among the changes that were made to increase the security.

I-B. Bluetooth Model

The Bluetooth protocol is introduced by the Bluetooth Special Interest Group (SIG) [7]. The protocol is standardized as IEEE 802.15 [24] for Wireless Personal Area Networks (WPAN). Bluetooth is already used in many applications despite some serious security concerns, e.g. [29]. Many manufacturers implemented Bluetooth poorly which sometimes enables an adversary to access private data of somebody's Bluetooth cell phone or PDA. In many implementations, the authentication is disabled to allow for an easier data exchange between mobile devices, such as the transfer of business cards between two PDAs. If the authentication is enabled, the 128 bit authentication key is derived from a PIN that was entered in all communicating devices. The length of the PIN varies between 8 to 128 bit. Note that in many cases the PIN is set to zero by default or is set short by the users because manually entering long PIN is not very user-friendly. The stream cipher E_0 was especially designed for the Bluetooth protocol. So far no efficient direct attack on E_0 is published but some high-order complexity attacks are presented in [29, 15], for instance.

We refer to the family of protocols that require users to enter their password, PIN, or key manually as the Bluetooth model. Solutions in the Bluetooth model do not scale well because the secret needs to be entered manually in each device.

I-C. The Resurrecting Duckling Model

Another symmetric approach is introduced by Stajano and Anderson in their resurrecting duckling model [51, 52]. The symmetric keys need to be exchanged over a secure side-channel prior to the execution of the authentication protocol. The authors suggest exchanging the symmetric keys by physical contact among the devices, for instance.

We refer to all protocols that require physical contact among devices in order to exchange their secret keys as a resurrecting duckling model. The requirement of physical contact among all communicating devices might be too restrictive in some applications.

I-D. Pairwise Key Pre-Distribution Model

Public key cryptography is not feasible in sensor networks and therefore only symmetric schemes are applicable. The approach that all sensors share the same secret key for authentication and encryption is not suited in sensor networks because sensors provide only weak physical protection. In this case, once an adversary gains physical access to a sensor in the network, she/he could read out the secret key, and thus, the entire network could be compromised. For this reason, sharing keys pair-wise seems to be a more reasonable approach. In addition, this approach enables entity authentication. Since sensors have very constrained memory, they cannot store symmetric keys of every other sensor in the network. To overcome this constraint, key pre-distribution protocols, which assign each sensor a subset of the total set of symmetric keys, are proposed recently. Note that the sensors of a network always belong to one domain. For most sensor networks applications, it can be assumed that a trusted authority can set-up all sensors before they are deployed. This process is called key pre-distribution.

Eschenauer and Gligor proposed a probabilistic key pre-distribution protocol in [18]. In their scheme, each sensor is initialized with a random subset of keys out of the entire key pool. When two sensors wish to securely communicate, they check if they directly share a secret key. If they do not, they have to try to find a common neighbor with whom they both share a key with and use this intermediate node(s) to establish a secure key.

In the pairwise key pre-distribution protocol proposed by Liu and Ning [38], the authors make use of the facts that most sensor networks are static, i.e. sensors do not move once deployed, and that the location of sensor can be predicted. They argue that each sensor has an expected location, thus, a sensor can be initialized with a set of keys from its expected neighbors. The authors argue that sensors can only talk to nodes in their direct neighborhood, because of their limited transmission range. By implementing a location-based approach, the probability that two neighbor nodes share a key is higher than in a probabilistic pre-distribution scheme. This approach is suited in static sensor networks, in which the location of single nodes can be predicted.

2.3 Hybrid Solutions

Some ad hoc network solutions combine symmetric and asymmetric crypto schemes to provide entity authentication and/or key establishment. We introduce two hybrid authentication models in the following section.

II-A. Password Model

Depending on the available memory size and the way the secret is exchanged, it might be desirable to share a short password instead of a long secret key. For instance, if the secret needs to be entered manually in all devices, a user-friendly password, such as a natural language phrase, could be used as the common secret. Note that such passwords are weak secret keys because they are prone to brute-force attacks due to their shortness, where user-friendly passwords are also prone to dictionary-attacks. If we want to use a shared password to derive a strong encryption key, we need to use an asymmetric crypto scheme. To implement this idea, we need a password-authenticated key exchange (PAKE) protocol that resists off-line dictionary attacks. Note that these protocols provide both entity authentication and the establishment of a session key. Due to the use of asymmetric crypto schemes, PAKE protocols require some heavy computational steps, thus, the computational costs of all PAKE protocols need to be examined with respect to their suitability in ad hoc network applications. We refer to the set of protocols that are based on this idea as protocols in the password model.

The combination of a weak password and an asymmetric crypto scheme to obtain a strong shared key was first introduced by Bellovin and Merrit [5]. They suggested using a password to encrypt a freshly picked short-term public key. One of their introduced variant is based on an encrypted DH key agreement. A PAKE protocol using human-memorable passwords was introduced in [31]. This protocol is, as far as we know, the only existing protocol of this kind which is proven to be secure in the standard model, i.e. without the random oracle assumption. In addition, the protocol does not require the presence of a trusted third party. Unfortunately, the protocol requires many heavy computations which is undesirable in ad hoc networks.

Asokan and Ginzboorg modified the DH-variant from [5] and introduced a multi-party PAKE protocol that establishes a session key in a contributory fashion among n parties by implementing an encrypted multi-party DH key exchange protocol [3]. The authors suggest the implementation of their protocol in a scenario where a group of people who meet at a conference and wish to spontaneously set up an ad hoc network. Due to the nature of the DH key agreement the protocol requires some heavy computational steps. Thus, the proposed protocol is only applicable in scenarios where the ad hoc devices are powerful enough to perform such computations.

II-B. Key Chain Model

Another hybrid approach to provide entity authentication is to use key chains. Using the elements of a hash chain for authentication was first introduced in 1981 by Lamport [35]. In hash chain schemes, a hash function $h(\cdot)$ is applied n times to a random value x. The initial value $x_0 = x$ is the so-called anchor and $x_n = h^n(x)$ is the last value of the hash chain. Each device computes its own hash chain, authentically exchanges x_n with its communication partners, and keeps the value x_0 secret. A device challenged by a value x_i from its hash chain can prove its identity by responding with the previous value x_{i-1} of the chain. Only a device that knows the anchor x_0 is able to compute the required response. Note that schemes implementing hash chains, also called key chains, provide only unidirectional authentication and no key is established during the protocol execution.

Weimerskirch and Westhoff proposed a protocol that requires neither the presence of a certification authority (CA) nor the use of certificates [57]. The computational costs are based on the computations of hash values and are thus very cheap. The anchor x_0 of the hash chain serves as the device's private key and the last value x_n as its public key. Since the introduced solution does not assume any secure channel for the exchange of the public keys, the public keys x_n cannot be exchanged authentically. The scheme provides only weak authentication which is more service than security oriented. The solution enables devices to recognize a device that has previously provided service. If the service was satisfying the node is able to identify the same node and request the same service again. Thus, the public key is bounded to a service and not to an identity.

In a later paper [58] the same authors strengthened the authentication for the price of two requirements: (1) temporary or permanent internet access; and (2) network devices with moderate computational power. In this scenario the public keys x_n of the devices are signed by a CA.

Therefore, at the time a device receives a public key, it needs to verify the CA's signature on the public key. For each communication partner, a device needs to perform one verification. Once the public keys are successfully verified, the scheme becomes the same as the original protocol and requires only cheaper computational steps.

2.4 Asymmetric Solutions

In the following paragraph, we describe different authentication models for ad hoc networks that are based on asymmetric encryption schemes. The public keys are used for entity authentication and for session key establishment. The session key is then used in a symmetric encryption scheme to provide confidential communication among the authenticated devices. The lack of a central CA is the main problem when implementing asymmetric protocols in networks without a fixed infrastructure. We distinguish four categories of asymmetric authentication models: (1) with CA and with use of certificates; (2) with CA and without the use of certificates; (3) without CA but with use of certificates; and (4) without CA and without certificates. The first category includes the distributed CA model; the second one includes the identity-based model and the self-certified public key; the third category contains the self-organization and the trusted subgroup model; and the fourth contains the certificateless public key model. In the following we will describe all these models and review some protocols that are proposed in them.

III-A. Distributed CA Model

In the distributed CA model the power of the CA is distributed to t network nodes by implementing a (t, n)-threshold scheme. The idea is based on the fact that a CA should not be represented by a single node, because nodes provide only weak physical protection and could be compromised relatively easily by an adversary. The approaches introduced in [59, 34, 39] are all based on the distributed CA model.

In 2001, Zhou and Haas introduced a protocol [59] which they claimed to be suited in networks without any infrastructure and consists of mobile hosts. Their idea is to distribute the power of the CA to $t+1$ special nodes, the so-called server nodes, that were present at the network initialization. The authors implement their idea by a $(t+1, n)$ threshold scheme. Any $t+1$ server nodes in the network are able to jointly issue certificates. Each member of the network is in possession of a private and public key pair. Members can request authentic copies of the public key of any communication partner from any group of the $t+1$ server nodes.

A node A needs to perform a *query* to obtain an authentic copy of B's public key. A initiates a *query* by broadcasting a request to at least $t+1$ server nodes. Each of the server nodes signs the requested public key with its share of the system's secret key. The $t+1$ partial signatures are then sent to a combiner node C, who combines all partial signatures and sends the full signature to A. Node A verifies the signature on B's public key and either accepts or rejects. The work load of the server and combiner nodes is tremendously heavy since they have to response to all submitted *queries*. The special role of the server and combiner nodes contradicts with the design goal of similar constrained devices. The protocol requires a fairly large memory for all server nodes because each server node needs to store the public keys of all network nodes. Instead of having server nodes sign the same public key many times, it seems to be more efficient if each node requests a certificate for its own public key from the server nodes and then return this certificate when requested by another node, as proposed in [34]. The protocol requires a large number of nodes to work efficiently. For all queries $t+1$ server nodes need to be in transmission range, which seems to be a restrictive assumption in general ad hoc networks.

The introduced procedure of an *update request* enables a node to change its own public key and inform the server nodes about the new key. How a node could authenticate itself to the server nodes is discussed neither for the first authentication nor for the update of the public key, especially since in the first authentication publishing a public key to the server nodes is fundamental for establishing trust in the network. The security of the system is based on this step and potential implementations, such as requiring physical contact or any other secure side-channel between the network node and the server nodes, are likely to be restrictive. The server nodes are responsible for keeping their public key databases updated that, again, adds computational load to the server nodes. The procedure of *share refreshing* prevents the system from being vulnerable to mobile adversaries and allows the parameters t and n of the threshold scheme in the running system to be changed. To refresh the secret key shares, each server node i needs to generate n new shares and encrypt each share s_{ij} with the public key of server j. In total we have n encryptions which requires n modular exponentiations in an RSA implementation or even more computational steps if we have a digital signature scheme that cannot immediately be used as encryption scheme. In the latter case, we need to establish a session key first. The computational costs of the share refreshing are too high if server nodes are conventual ad hoc devices.

Kong, Zerfos et. al. proposed a similar approach in [34]. The authors presented a protocol that combines the RSA protocol with a threshold scheme. They extended the tasks of the CA, which is presented by k nodes here, to issuing, renewing, and revoking certificates. Note that there are no special (server) nodes in this implementation as required in the previously discussed solution. Hence, all nodes have equal roles. The verification of certificates requires less computational and communication overhead than in the previous protocol because no devices other than the ones that wish to communicate are involved. The processes of issuing, renewing and revoking certificates require again the interaction of many devices, here at least $k + 1$, and are thus not very efficient. To obtain a certificate, a device needs to identify itself to K nodes. The authors suggest that the identification process is performed by physical contact of the devices or over any other secure side-channel. This seems to be a very restrictive assumption since it requires K nodes in the direct neighborhood. Also the combining of all partial signatures to obtain the full signature, i.e. the public key certificate, requires some heavy computations. The protocol performance is already analyzed by the authors themselves [34]. Their results are based on implementation on platforms, such as a Pentium III/500 laptop. Our design goal is an efficient protocol on more constrained platforms, such as a PDA or a cell phone.

In the extended version of the protocol [39], shares can be updated in case compromised nodes are detected. Nodes are notified about compromised nodes by flooding a list of the revoked certificates. Another novelty is that the parameter k can be changed in the running system. Thus k can be adjusted according to the present network state, e.g. the number of present nodes which makes the solution more flexible. In the extended protocol new certificates can only be issued and distributed by a centralized CA. Once a node is in possession of a certificate, it can request a renewal, revocation, or the public key of other nodes from k nodes. This assures that every holder of a certificate has successfully authenticated itself to a trusted party at least once. This approach requires that each node to have contacted a central CA and requested a certificate before joining the network. This can be restrictive in some scenarios where a CA might not be available all the time.

All three previously discussed protocols in the distributed CA model are one of the first papers that consider the special features of ad hoc networks, such as the lack of infrastructure and the mobility of the user. The authors suggest the utilization of threshold schemes to provide fault tolerance, which is a desired property in sensor networks. However, all these solutions do not take into account that ad hoc network devices are constrained in computational and communication power. We can observe that protocols in the distributed CA model require some heavy computations and a large computational and communication overhead. In addition, the discussed solutions all require a fairly large number of network nodes to be present, which is not necessarily given in all ad hoc network applications at all times.

III-B. Identity-Based Model

Identity (ID)-based schemes, introduced by Shamir in 1984 [49], do not require any key exchange prior to the actual authentication, because common information is used as the public key and the certificate at the same time. ID-based cryptography schemes are based on the idea to use human readable (unique) identities, such as names, email addresses, etc., as public key. Thus, the identities are self-certifying, e.g. Alice's public key and certificate could be $P_A = Alice@uwaterloo.ca$. There are two main advantages of using ID-based systems. First, no public key certificates are required, and second, no exchange of the public keys is required. Implementing the revocation of the public keys is easy in such systems and might be achieved by adding an expiry date to the public key, e.g. $P_A = Alice\|march04$. ID-based schemes require a CA at the initial stage of the network in order to generate and distribute the personal secret keys of all users. After that phase the CA becomes redundant. The fact that the CA knows the secret keys of all users is generally considered as drawback of ID-based schemes. The confidential and authentic channel between the CA and each network device required for the distribution of the secret keys is another drawback. Note that in other asymmetric scheme, an authentic channel is sufficient, because only public data is transmitted. Due to its knowledge of all secret keys the CA is a key escrow. If desired, the power of the CA could be limited by one of the following approaches: (1) assigning an expiration date to the system's master secret; (2) encrypting all messages using additional private/public key pairs which are unknown by the CA [16]; or (3) distributing the power by implementing a threshold scheme, requiring k nodes to perform all tasks [32].

Khalili, et. al proposed a protocol for key management and authentication in ad hoc networks that is based on an ID-based scheme in [32]. They suggest combining an ID-based scheme with a (t, n)-threshold scheme to overcome the requirement of a centralized CA. This implementation also reduces the power of the CA by distributing the power to t network nodes. The authors do not provide an actual protocol and there are many open questions for an implementation. For instance, how a new node A could receive its secret key is not discussed. The t nodes representing the CA could each compute a share of A's secret key and send it to A. In this case, an eavesdropper could easily obtain all shares and decrypt all subsequent messages that are encrypted by A's identity or he/she could sign messages using A's secret key to impersonate A. Thus, additional assumptions need to be made such as a secure channel for the key distribution. Another open problem is the computation of the system's master key. The authors assume that the key is computed in a distributed fashion by the n nodes that were present at the time of the network initialization. They do not explain how this can be implemented by using existing ID-based crypto systems although this function is crucial for the system. Note that if the master key of the system is compromised the entire system is compromised.

III-C. Self-Certified Public Key Model

In schemes with self-certified public keys, the certificates are embedded in the public keys themselves. The identity of a user is part of his/her public key. Note that other than in ID-based schemes the identity itself is not directly used as a key. Hence, the public keys need to be exchanged prior to the communication. The authenticity of the public keys is provided by the keys themselves. We do not need certificates or other mechanism to provide an authentic channel. This approach helps to save some bandwidth and memory space, because certificates do not need to be transmitted and stored. In this model, each device possesses a private and public key. A CA is required to issue the self-certified public keys. The CA generates the self-certified public keys using the device's public key, identifier, and the CA's secret key as input. Note that the CA does not know the secret keys of the devices. The network devices use their self-certified public keys for all authentications in the network. Encryption and signing in self-certified schemes are different from regular asymmetric schemes because the secret and the self-certified public key do not directly correspond with each other.

In 1991, Marc Girault introduced the concept of self-certified public keys [17]. In his approach the CA issues self-certified public keys to all devices. The users need the CA's public key to verify the authenticity of a public key. Girault presents an authentication protocol for his scheme, but as mentioned before, in most ad hoc networks it is desired that a session key is established after a successful authentication. Girault also presents a key agreement protocol which is based on a DH key agreement protocol. The proposed protocol is resistant to a man-in-the-middle-attack because of the use of self-certified public keys. However, this protocol cannot be used to establish a session key because the long-term public keys are used to derive the common key and thus yield to the same shared key every time the protocol is executed between the same two parties. A new short-term self-certified public key would be needed for every protocol execution. This is not practical because generating a self-certified public key requires the presence of a CA, which is not available all the time in ad hoc networks. We can conclude that this approach is not applicable in ad hoc networks and a suitable solution using self-certified public keys in such networks has not been proposed yet.

III-D. Self-Organization Model

The self-organization model emphasizes the self-organization property which is a unique and challenging feature of ad hoc networks. Network nodes issue and distribute their own certificates. Nodes also sign other certificates. The model assumes the existence of trust between some nodes and generates trust between nodes in a PGP manner.

In 2001, Hubaux, Buttán, and Čapkun introduced a protocol in the self-organization model [28]. In [13], the same group of authors extended their ideas, where the new feature is that certificate revocation is provided. In their approach, every node A has a public and private key and holds a list of certificates of all nodes that she trusts (*out-bound list*) and a list of certificates of all nodes that trust her (*in-bound list*). In order to verify a certificate, nodes try to find a trusted path between them by merging their in-bound and out-bound lists. In the best case, A needs to verify only B's (in-bound) certificates on the path. In the worst case, A needs to verify all certificates on the trusted path except the first one that she issued herself. Consequently, the performance of pre-authentication highly depends on the length of the trusted path. For this reason, the authors introduced an algorithm for efficiently finding the shortest path. According to their simulation results in [13], A would need to perform 4 verifications in the best case or 6 verifications in the worst case if we use a PGP graph of size 24. Obviously, the results highly depend on the network structure, i.e. the number of nodes and certificates, and are hard to generalize or predict for arbitrary networks. However, A probably needs to verify more than 1 certificate and thus needs to perform more verifications than in the previously discussed protocols in the distributed CA model. But other than protocols that are based on threshold schemes, this approach is cheap in the set-up phase and does not require any heavy computations from any parties other than the ones that wish to communicate.

We believe that there might be some unsolved security problems when merging the certificate lists because both parties send their list unprotected and unauthenticated over an insecure channel. This might enable vulnerabilities, such as a man-in-the-middle-attack.

III-E. Trusted Subgroup Model

In this model, all members of a subgroup trust each other. If two nodes wish to authenticate each other, the subgroups search for intersections to create a trusted path. The solution requires that at least some nodes trust each other a priori. Furthermore, the solution requires a large number of nodes or subgroups and relationships among them.

The authors Gokhale and Dasgupta introduced a solution that is based on existing trust in small groups of nodes, which they called troups [19]. Each troup has a troup controller which plays a special role in the troup. The special role of troup controllers is an undesired requirement in ad hoc networks because ad hoc devices have preferable equal roles and similar constraints. The protocol description is very rough and leaves many questions open. The proposed solution requires many modular exponentiation, which makes the protocol undesirable for an implementation in ad hoc networks. We believe that this approach is limited to a few selected applications. The author provide some simulation results. Their performance measurements were done on a Pentium III which is much more powerful than most ad hoc devices. Therefore, the results are not very meaningful in an ad hoc network context.

III-F. Certificateless Public Key Model

Another family of asymmetric approaches comprises all protocols that use public keys without certificates, i.e. neither keys or identifiers nor embedded or separate certificates are used as certificates. In this model, devices exchange their public keys over an authentic channel where the authenticity is achieved by visual or physical contact among the communicating devices. This usually requires that all participants are located in the same room. In addition, all participants must already trust each other a priori. Consequently, this approach is well suited in all scenarios where users trust each other and are located close to each other. The approach is not applicable in any other scenario. In all cases where devices can perform physical contact, implementing symmetric schemes seems to be more reasonable.

Balfanz et. al introduced a protocol where all public keys are directly exchanged over an location-limited channel [4]. The short distance between two devices ensures authenticity. Since the communication channel is still prone to eavesdropping, the authors suggest exchanging public keys. Due to the authentic channel, the use of certificate is redundant. The public keys can be directly used for authentication and/or encryption.

Performance

A comparison of the network performance of different asymmetric design choices - centralized, e.g. [59], peer-to-peer, e.g. [28], and localized, e.g. [34], is presented in [12]. Their result shows clearly that localized implementations provide the best network performance among the considered solutions.

3 Definitions and Properties

Due to the diversity of applications, there is still a lot of confusion about the definition of ad hoc networks and their properties. In this section, we point out what distinguishes ad hoc networks from other network types. We then define the properties of ad hoc networks and their devices.

3.1 Ad Hoc Networks vs. Wireless Mobile Networks

Sometimes ad hoc networks are mistaken for wireless mobile networks. The main difference between both networks is in their infrastructure. In contrast to wireless mobile networks ad hoc networks do not rely on fixed infrastructures as stated in [59]. The Latin expression *ad hoc* stands for *formed for* or *concerned with one specific purpose*. Using this translation we could say that ad hoc networks are instantly formed to serve a special purpose. This definition implies a dynamic infrastructure, because the network is formed at the moment the use of a service is required and ceases to exist after the network fulfill its purpose. Ad hoc networks can be the extension of any other network with fixed infrastructure.

Another difference between wireless mobile and ad hoc networks are their network devices. All devices of an ad hoc network are likely to have similar constraints, e.g., regarding their computational power. Consequently, we cannot implement protocols in ad hoc networks which require only one device to perform expensive computations, whereas the other device carries out the cheap operations. This type of unbalanced protocols, e.g. realized in RSA with small exponent, are used in many wireless mobile networks with master-slave architecture. Ad hoc networks require a protocol where all devices perform equally heavy and many computational steps.

3.2 Ad Hoc Networks vs. Peer-to-Peer Networks

Peer-to-peer (P2P) networks consist out of two or more nodes and can be formed instantly without the help of a central coordination. P2P devices usually have the same capabilities, and they use existing networked structures, such as the internet, to communicate with each other. The most common application of P2P networks is file sharing. This can be implemented by companies to enable file sharing among employees without the presence of a server. Another popular application allows that internet user directly share files from their hard drives as provided, for instance, by Napster and Gnutella. The properties of P2P networks make them ad hoc networks. But not all P2P networks are ad hoc network because not all ad hoc network implementation utilize an existing structure for the communication among devices. Also the variety of applications of ad hoc networks is beyond the scope of P2P networks.

3.3 Ad Hoc Computing vs. Pervasive Computing

Pervasive computing, also called ubiquitous computing, is often described as a new computer era. Many people predict that this technology will be used everywhere and by everyone in the near future [53]. The devices are usually very small and can be embedded in any type of objects, as for instance, fridges and other home appliances. Users are sometimes not even aware of the existence of

Table 2: Authentication models for ad hoc networks

Encryption scheme	Model	Implementation
I. Symmetric	A. IEEE 802.11 (b) Model	No authentication or keys exchanged over secure side-channel outside the IT system [24]
	B. Bluetooth Model	PIN manually entered in all devices [7]
	C. Resurrecting Duckling Model	Key exchanged by physical contact [51]
	D. Pairwise Key Pre-Distribution Model	Sensors are initialized with subset of key pool before deployed; random subset [18], subset based on expected location [38]
II. Hybrid	A. Password Model	Shared password is used for authentication and securely establishing a session key [3]
	B. Key Chain Model	Anchor x_0 of hash chain serves as private key and x_n as the public key [57, 58]
III. Asymmetric	A. Distributed CA Model	CA represented by n special server nodes using a threshold scheme [59], Any K nodes represent the CA using a threshold scheme [34, 39]
	B. Identity-Based Model	Identity used as public key and no certificates are required; CA distributed by using threshold scheme [32]
	C. Self-Certified Public Key Model	Certificate is embedded in public key [17]
	D. Self-Organization Model	Trusted path between 2 nodes, idea similar to PGP [28], or more advanced [13]
	E. Trusted Subgroup Model	Small groups of nodes that trust each other a priori, building trusted path by joining subgroups [19]
	F. Public Key without Certificate Model	Public key distributed over short range (visual or physical contact) [4]

the embedded electronic chips. Typical ad hoc network devices are for instance PDAs, cell phones, laptops, and sensors. Thus, users are aware of the device when they use them, which is the main difference between the two network technologies. Perhaps devices used for pervasive computing can form an ad hoc network, but ad hoc devices are not necessarily embedded devices.

3.4 Categories

We divide ad hoc networks into two categories, namely *mobile ad hoc networks (MANETs)* and *smart sensor networks*. These two categories were introduced by the National Institute of Standards and Technology (NIST) [42]. The application areas, the security requirements and the constraints of the single devices differ strongly for these two realizations of ad hoc networks. MANETs are of great interest for civil usage and consequently the devices are desired to be cheap, lightweight, and easy to use. Whereby the research on smart sensor networks is mainly driven by the military. Hence, security is the target objective in sensor networks, independent of the development and manufacturing costs of single sensors. We can deduce from the applications listed in Table 1 that civil applications and health applications usually make use of MANETs, whereas military and governmental services utilize smart sensor networks. Typical devices of MANETs are PDAs, laptops, cell phones, etc., and the devices of smart sensor networks are sensors. Usually sensors are densely deployed. For this reason the number of nodes in a smart sensor network is usually over an order of magnitude higher than the number of nodes in a MANET. Sensor are even more constrained than typical MANET devices and prone to failure. We will clearly distinguish between these two types of ad hoc networks in the following section.

3.5 Properties

We define the properties of ad hoc networks as follows and describe them briefly.

Temporary Network: As mentioned before, an ad hoc network is formed to fulfill a purpose and it ceases to exist after fulfilling this purpose. Consequently, the network exists only for a limited time period which we denote as a temporary network.

Dynamic Network Architecture: Nodes can arbitrarily join or leave the network. Thus, the network architecture might change frequently.

Short-range Network: Most ad hoc networks are wireless networks using infrared or radio frequency for transmission. As a consequence, the transmission range is limited. For instance, IrDA Data protocols of the Infrared Data Association [26] have a typical transmission range of 2 meters between two devices. The range is usually shorter for low power devices, where it typically varies between 20-30 cm. The range can be increased by sending packets to neighbor nodes that are within the transmission range. These neighbor nodes will forward the packets until they reach their destination. These kind of networks are called multi-hop networks.

Self-organizing Network: This property is unique to ad hoc networks and distinguishes them from all other network types. After the network initialization, the network should be self-organized. For instance, if network nodes join or leave the network, the other nodes carry out all required steps independent of a server or any other third party. These steps could include distributing keys or other

Table 3: Comparison of capabilities of a SmartDust sensor and a conventual PDA

	SmartDust sensor	PDA
Processor	eg. ATMEL 90LS8535 processor32 8-bit registers	32-bit Intel 386 Processor
Memory	8Kb programm memory 512 bytes data memory	2 MB SRAM 16 MB flash

data, and establishing shared secrets. Consequently, no external trusted third party is involved in any network activities after the network has been set up.

Constrained Devices: Another characteristic of ad hoc networks are their constrained network devices, which makes implementing any security a difficult task. The constraints of ad hoc network devices are small CPU, small memory, small bandwidth, and limited battery power, as first summarized in [51]. The devices have only weak physical protection. If an adversary has access to the device, it is most likely that he/she can read out all data. Thus, an adversary could gain access to confidential data such as secret keys. In Table 3, we show some typical capabilities of a SmartDust sensor node and a standard PDA as representatives of a sensor and a MANET device, respectively. The comparison of both devices illustrates that the computational and communication capacities of sensor nodes are over an order of magnitude lower than the ones of MANET devices. In addition, the battery power of both devices differ significantly. We list the power sources and ranges of some typical ad hoc network devices in Table 4. The table demonstrates that the power resources of sensors are very constrained, which limits the sensors' computational and communication capabilities significantly. We observe that the power resources of typical MANET devices are usually stronger than the ones of sensors but nevertheless still constraint. Note that batteries of MANET devices are likely to be rechargeable in most applications, whereas the batteries of most sensors cannot be recharged once released. A new trend in sensor technology is the use of energy scavengers instead of conventional batteries. Scavengers can convert noise, heat, vibrations, or light from the environment into electrical power. Sensors that use such scavengers are totally independent because they do not need to be recharged once deployed, and are thus not required to be accessible anymore.

Similar Devices: In ad hoc networks all devices have similar constraints. This distinguishes the architecture of an ad hoc network from a client-sever structure. In client-server networks some heavy computations can be shifted to the server, which is computationally stronger than the clients. In contrast, all computations in an ad hoc networks should be balanced among all participants. It should be obvious that protocols using balanced computations can be easily adapted to server-client networks but imbalanced protocols cannot be used in ad hoc networks.

3.6 Network Phases

We introduce two network phases for ad hoc networks, namely the *network initialization phase* and the *running system phase*. In the first phase, the nodes that are present at the time the network is formed are initialized. The self-organization property of the network is sometimes not required at this stage, i.e., a CA or any other Trusted Third Party (TTP) might be available in order to

Table 4: Comparison of power sources and ranges of some ad hoc network devices

Device	Power Source	Power Range
Desktop Computer	Power Grid	150W -500W
Laptops	High Capacity Battery	10W-120W
PDAs, Cell Phones, embedded electronic chips	Battery	100mW-10W
Smart Sensors	Tiny Batteries	1mW-100mW
Smart Dust, RFIDs	Energy Scavenging	1 μW- 500 μW

initialize the nodes with the required data. This phase is not mandatory in all ad hoc networks. After the initialization phase (if there is any), nodes might frequently join or leave the network in the running system phase. It is desirable for ad hoc networks to be self-organized at this stage.

3.7 Authentication Phases

We distinguish two authentication phases for the authentication of any network node. The first phase is executed to exchange the data that is required in all later authentications between the same devices. This phase is called *Imprinting* in the Duckling Model [51], and *Initialization* in the Bluetooth protocol [7]. Henceforth we will adopt the term *Pre-Authentication* from [4]. The data that is exchanged in the pre-authentication phase needs to be sent over a secure channel, where a secure channel refers to an authentic and confidential channel for exchanging symmetric key data, and an authentic channel for exchanging public keys or other public data in asymmetric schemes. Pre-authentication is not to be confused with the previously described *network initialization.* The pre-authentication phase is required for each network device as long as a device wants to communicate securely with any other network device. Pre-authentication is not limited to the devices present at the time of the network initialization, but also nodes that subsequently join the ad hoc network need to obtain all shared data and required key material, even though the network environment might have changed, e.g. when a CA is not present any longer. During the second phase, the authentication phase, the actual authentication is executed over an insecure channel using the authentic data that was exchanged in the first phase.

4 Implementation Parameters and Protocol Framework

In this section we define some ad hoc network parameters. We then define the design goals that all ad hoc network protocols should meet. Finally, we introduce two different protocol frameworks for providing authentication and secure key establishment in ad hoc networks at the end of the section.

4.1 Parameters

The proper choice of parameters in each implementation is important in order to achieve the security objectives of a particular application and to keep the solution applicable at the same time. Other than the properties of ad hoc networks, which are inherent, a set of parameters can be chosen for each implementation. The choice of some parameters might be mandatory in particular scenarios and some parameters might depend on each other. We list some potential parameters below.

Mutual Entity Authentication vs. Broadcast Authentication: Before developing or implementing protocols, we should be aware of the kind of authentication we need in our application. Do we need broadcast authentication or entity authentication? In all broadcast communication systems, each data packet that is sent from a source reaches a number of receivers. In those communication systems, broadcast authentication, also called source authentication, is of great importance. Broadcast authentication enables all receivers to verify if the received data was really sent by the claimed source and whether it was modified en route. In sensor networks, broadcasting is the main type of communication. Sensor networks used in military scenarios are examples of security sensitive applications. In a hostile environment, all sensor nodes need to be able to verify if the received commands were sent by their own base station and not by the enemy. Boneh et. al showed in [8] that a protocol which provides *compact collusion resistant broadcast authentication* needs to rely on either digital signature or time synchronization. Due to the resource constraints of sensors the latter approach seems to be the only applicable method. In 2000, Perrig et. al introduced their *Timed efficient Stream Loss-tolerant Authentication (TESLA)* protocol [44] that provides broadcast authentication based on time synchronization. The TESLA protocol and its successors [46, 45, 37] are all based on time synchronization and use only symmetric cryptographic function. These two features make the protocols interesting for an implementation in sensor networks. The "micro" version of the TESLA protocol, called μTESLA [45], and its modified version [37] are especially designed for a use in sensor networks.

Mutual entity authentication provides the online authentication between two devices. After successfully authenticating each other, two devices usually establish a session key that they use to encrypt all further communications. This kind of authentication enables a secure channel between pairs of devices, whereas broadcast authentication enables a secure channel for one-to-many communications. Sometimes it might be sufficient to prove that a device belongs to a particular group, and the single nodes do not need to be distinguishable. For instance, in most sensor networks, all sensor nodes send information back to the base station. The base station usually does not distinguish between the single nodes, as long as the message is sent by an authentic sensor. Throughout this report we only consider mutual authentication of devices.

MANET vs. Smart Sensor Networks: As discussed earlier in this report, MANETs and smart sensor networks have different properties and security requirements. We have to consider the characteristic of the respective network when developing or implementing protocols. Consequently, suited protocols differ for both kind of networks and they need to be developed or be chosen according to the network type.

Wireless vs. Wired: Wireless networks are prone to eavesdropping. Furthermore, their communication bandwidth is more restricted compared to wired networks. It follows that the security requirements in wireless networks are usually more challenging to implement than in wired ones. We should always assume a wireless communication channel when evaluating or developing ad hoc network protocols, because all protocols that are secure in wireless environments can be securely adopted to wired networks but not vice versa.

One-hop vs. Multi-hop Network: To extend the communication range of single nodes, most networks support multi-hop communications. In a multi-hop network, every node acts as a router and forwards packets until they reach their final destination. Therefore, routing protocols that deal with the dynamic structure of ad hoc networks are required. In this report, we focus on point-to-point authentication, thus we consider only one-hop scenarios. We do not take aspects of routing and its associated problems into account. In the case of multi-hop networks we assume that all intermediate nodes correctly transfer the message to the recipient and consider intermediate nodes as a part of the communication channel.

Hierarchical vs. Flat Topology: Hierarchical ad hoc networks haven been proposed as alternative to flat ad hoc topologies to overcome some limitations of the latter, as for instance described in [9]. In a hierarchical ad hoc network we have several layers each consists of a set of similar devices. For instance, the lowest layer consists of the least powerful devices, e.g. sensors, and each level above consist of some more powerful devices, where the top level could be the internet. In this way, all heavy computations could be shifted from the very constrained devices to the more powerful ones and thus asymmetric schemes could become feasible. For this reason, the model is attractive for sensor networks. The question to ask is how reasonable is the assumption of the accessability of higher layers by all sensor networks. In most military applications, this cannot be assumed and the practicability of this approach needs to be further analyzed for other ad hoc applications.

Controlled vs. Uncontrolled: Stajano and Anderson [51] were among the first to consider the special properties of ad hoc networks, they assumed a controller (mother duck) and several devices that are controlled (ducklings) in a typical ad hoc network. In their resurrecting duckling model, the mother duck imprints their ducklings, who, from then on, follow their mother. In another more recent paper Messerges et. al [41] described some applications that require a controller, e.g. sensor networks used for industrial control and building automation. In networks without a controller all nodes have similar roles and are assumed to have similar resource constraints. Whether we have an ad hoc network with or without controller depends on the application. Note that it is usually harder to develop and implement a protocol for ad hoc networks that consist of similar devices with equal roles in the network.

One Network Phase vs. Two Network Phases: In some scenarios, it might be reasonable to assume that a trusted third party is present to initialize the present devices, whereas in some other scenarios it might not be. In the first case, we have a network initialization phase, in the latter one we do not. The absence of a network initialization phase is very challenging since none of the device could share information with any other node at the time the network is formed.

One Domain vs. Multiple Domains: All devices in one domain share the same domain parameters. Domain parameters could be a shared key that has been distributed during the network initialization, a certificate issued by the domain's CA, or system parameters required for some computations. In most sensor networks, it is reasonable to assume one domain. For instance, in a battlefield scenario, all sensors belong to the same army and can be initialized by the commanding unit before they are deployed. In many MANETs, the devices are from different domains. Providing authentication in those scenarios is harder to implement and both parties have to agree on some common system parameters. A main problem is caused by certificates issued by different CAs that are not cross-certified. How can a device verify a certificate issued by a CA other than its own without having access to the internet or another backbone? This and other questions have to be considered when implementing an authentication protocol in ad hoc networks where devices are not all from the same domain.

Mobile vs. Static Nodes: Although it is widely believed that all devices in ad hoc networks are mobile, we have observed that the location of the devices are static in many applications. For instance, in most sensor networks the sensors usually do not move once deployed. In all cases with mobile nodes we have to take the effects of the mobility into account and implement the protocols accordingly.

Availability of Trusted Third Party: Before implementing public key solutions, we need to consider the availability of a CA or any other TTP. We distinguish four different cases.

1. **CA always available**

   The case that a CA is always accessible by all network nodes is generally not considered as an option in ad hoc networks, because ad hoc networks should be self-organized after their initialization. If a CA is permanently available we could implement solutions that require certificates or implement Kerberos-like solutions where the TTP distributes session keys. However, in the future it might be reasonable to assume internet connection availability in ad hoc networks. In this case we only need to cope with the resource constraints and mobility of the devices.

2. **CA available at network initialization phase and every time a node joins**

   The second option comprises all scenarios where a CA is available to issue certificates, and generate and distribute key material and system parameters at the initial stage of the network. The CA is also available for all nodes that subsequently join the network in order to obtain the required system parameters and keys. The assumption that a CA is available every time a new node joins the network is not as restrictive as it might sound. The CA does not need to be accessible by all network nodes every time a new node joins a network. There could be implementations in which nodes contact a CA in order to receive the required data, such as a certificate of the public key or a symmetric key, before joining the network.

3. **CA available at network initialization phase**

   This option is similar to the previous one, with the difference that subsequently added nodes cannot access the CA. After the initialization phase, the CA cannot be contacted anymore by any of the nodes, including the nodes in the networks and newly joining nodes. Usually this is called the self-organization property of the network. The present network nodes are responsible to take over the tasks of the CA, such as issuing, renewing, and revoking certificates.

4. **No CA available at any network phase**

   If no CA is available at all and we still want to use public key encryption schemes, the nodes need to issue their own certificates or we need to implement a model that does not require any public key certificates. The first case can be realized by protocols in the self-organization model and the latter case by protocols in the certificateless public key model. Please refer to Section 2.4 for the description of the models.

Security of Communication Channel: We distinguish two communication channels. One channel to exchange the data that is used for all later authentications during the pre-authentication phase and another channel to execute the authentication and key exchange protocol.

1. **Channel for Pre-Authentication**

    - **authentic and confidential**
      If we have a *confidential and authentic* channel for pre-authentication, we can securely transmit secret key data, such as symmetric keys or passwords. This channel can be outside of the IT-system, e.g., all devices are set up with a secret during their manufacturing. In this case, all potential communication partners would need to be in the same domain. Another approach to share a secret outside the communication system is introduced as a conference model by Asokan and Ginzboorg [3]. Here, the secret is written on a blackboard in a conference room. In the Bluetooth protocol [7] users manually enter a PIN in each device. The secret could also be transmitted by physical contact of the devices as described by Stajano and Anderson in their resurrecting duckling model [51]. If visual contact of the devices is provided, the keys could be transmitted over a wireless location-limited channel. Note that devices need to trust each other a priori in those scenarios. The limited distance of the devices ensures that the channel is authentic since we receive the keys from a particular device and not an attacker in the middle. Confidentiality is achieved by limiting the transmission range to a short distance, e.g. within a room by using infrared.
    - **authentic**
      If our system provides a channel that is just *authentic*, only authentic data can be exchanged, as for instance public key material or other public system parameters. In contrast to transmitting confidential data, we do not need to protect authentic data from eavesdropping. Note that eavesdropping poses a serious threat in all wireless channels. A common method to establish an authentic channels is the use of public key certificates. Implementing this solution requires an entire infrastructure for issuing, distributing, and desirably revoking certificates. Another method to provide an authentic communication channel is by physical or visual contact among the devices that wish to communicate.
    - **other**
      If the channel can neither provide confidentiality nor authenticity, or it provides confidentiality only, no key data or other information can be exchanged before the authentication starts. For those applications, we need a solution which does not require any pre-authentication. We believe that using ID-based crypto systems or protocols using self-certified public keys could be a suitable approach to overcome this problem since a pre-authentication is not required in those systems.

2. **Channel for Communication**

    The actual communication channel between all network devices is always considered to be insecure. A secure channel is established by using the information exchanged during the pre-authentication phase and/or the device's initialization, where secure means authentic and/or confidential. We do not consider communication channels that only provide data confidentiality since talking confidentially to somebody without knowing if we are talking to the right person makes no sense at all. We do not consider ad hoc network applications that do not require any secure communication, such as the "talking" museum items of HP's cooltown [27], or beaming business cards from one PDA to another. These kind of applications do not need to implement security such as authentication and key exchange protocols.

Level of Resource Constrictions: The level of constrictions of the network devices is determined by the application. In sensor network applications, for instance, the network consist of devices which are very constrained. Depending on the computational constrictions of the devices it might be feasible or infeasible to execute protocols requiring heavy computations, such as modular exponentiations. In addition to the computational constraints, we have to consider the communication constraints when designing the protocol. Furthermore, the level of limitation of the nodes battery power needs to be considered. We have given some examples of the power ranges of some typical ad hoc devices in Table 4 earlier in this report.

We believe that in the future, MANET devices are becoming more and more powerful and will be able to perform modular exponentiations and other complex computations. The number of heavy computations should still remain small since these computations require much power and the battery power is very limited. For the same reason the number of exchanged messages should remain limited, because sending messages also requires a lot power.

Location Awareness of Devices: If devices can provide information about their location, such as their geographical coordinates, the additional data could be used for their authentication. Consider the following scenario, you have visual contact with another user and he/she provides you with his/her authenticated GPS coordinates. When combined, they can provide an authentic channel which can be used to exchange public key data. Instead of visual contact, you might know the approximate location of your communication partner which serves the same purpose. In static networks, the location of devices might be predictable. For instance, in some sensor networks, the sensors have an expected location. This fact can be used for authentication of the devices and is implemented in a location-based pairwise key establishment protocol [38], for instance.

To be able to provide the present location of mobile devices, an additional integrated chip is required, such as a GPS chip. Some high-end PDAs are already equipped with a GPS chip. There are many different systems that provide location coordinates depending on the network range and location. The most commonly known systems for tracking down devices are satellite navigation systems, such as GPS, or the European equivalent Galileo. There are also systems for locating devices inside a building using different communication channels, such as visual, ultra sonic, radio, or infrared channels. The latter is implemented and currently used at four sites in Cambridge, England using the *active badge* [55] system. Another class is the network based positioning system, e.g., GSM, and WLAN. By using GSM, a device's cell could be determined, whereby the accuracy depends on the cell size.

Symmetric vs. Asymmetric Encryption Schemes: Both encryption schemes can be implemented in MANETs, but implementing asymmetric schemes is infeasible in sensor networks. If we wish to use asymmetric crypto systems in MANETs, the protocols should only require very few heavy computational steps. See the design goals in Section 4.2 for more details. Due to their cheap computational costs, symmetric schemes seem to be well suited for sensor networks and some MANET applications. Examples of suited MANET applications are most PANs, the conference scenario, and all other applications with a limited number of network devices in which all nodes are located within a limited range.

A Priori vs. No Key Sharing: Using symmetric encryption requires a shared secret before the authentication protocol is executed. Asymmetric schemes also require some pre-shared information which we call system parameters. System parameters are public and hence do not need to be exchanged and stored confidentially. Whether public keys need to be exchanged prior to the execution of the authentication depends on the underlying authentication model (see Section 2.4).

4.2 Design Goals

We now derive the design goals that all ad hoc network protocols should meet in order to be applicable. All ad hoc network protocols should be developed according to these design goals. We would like to mention that trust cannot be generated among nodes by the execution of protocols. Therefore, we need to trust the claimed identity beforehand. Authentication protocols verify if the claimed identity and the data and/or key material belong to the same source. If authentication is successful, we accept the data, key, etc., because we trust the verified identity. We will discuss the requirements of ad hoc networks only. Please refer to [40] for all general design goals of authentication protocols. We separately present the design goals for MANETs and sensor networks in the following section.

4.2.1 MANETs

Few computational steps: Due to the limited battery power of all ad hoc devices, desirable protocols require as few computational steps as possible. Too many computational steps would drain the battery.

Balanced computational steps: We assume that all ad hoc network devices have similar constraints. For this reason, a suited protocol should be balanced, i.e. all devices need to perform approximately the same number of equally heavy computations.

Cheap computational steps: Due to the limited computational power of ad hoc devices, preferable protocols should mainly require cheap computations. However, the processors of most ad hoc devices, such as PDAs, are becoming more and more powerful, and therefore heavy computations, such as modular exponentiations, are becoming feasible. Since heavy computations require more battery power, it is important to restrict the number of heavier computations.

Few message flows: Since the transmission of messages requires a lot battery power, the number of exchanged messages is desired to be as small as possible.

Small messages: Due to the nature of wireless networks, the communication bandwidth is very small. If messages are too large, they will be split into several packets. Sending many packets contradicts with the previous design goal, therefore small data packages are desirable.

Small program memory requirement: Because of the constrained memory of all devices, protocols should not require much memory space.

Small data memory requirement: Due to the very limited data memory, protocols should not require the storage of many system parameters and keys. Hence, small keys and system parameters are desirable.

Restrict consequences of data disclosure: MANET devices provide a low level of physical protection only, thus, once an attacker gains access to the device, he/she is usually able to obtain the stored data, including the key material. Note that this attack is quite reasonable since such devices cannot be protected as some servers are locked away in secure rooms, for instance. The protocol should be designed in a way that the disclosure of the stored data does not compromise the entire system. When all devices share the same symmetric key, the entire system is compromised if one key is revealed. Thus, solutions using different keys for different devices or communication partners are desirable. Also the possibility of how such a disclosure can be detected within the system needs to be considered when designing a protocol.

4.2.2 Smart Sensor Networks

Few computational steps: For the same reasons as those for MANETs, the number of computational steps in a protocol should remain small. In contrast to MANETs the batteries of sensors cannot be re-charged in most sensor network applications and is much more constrained as we demonstrated in Table 4. Thus, in sensor networks, we have to conserve the battery power of all nodes as much as possible.

Cheap computational steps: The computational power of sensor nodes is significantly lower than the one of MANET devices. We cannot assume that sensors are able to perform any kind of heavy computations, e.g. modular exponentiations. Thus, we cannot implement asymmetric schemes in sensor networks. Suitable solutions must be purely based on symmetric cryptographic primitives that require only cheap computations.

Few message flows: The same arguments for MANETs can be applied. Due to the even more constrained battery power and the fact that batteries cannot be re-charged in some scenarios, this design goal is even more restrictive for sensor networks than for MANETs.

Small messages: Same arguments as in the previous item.

Small data memory requirement: The same reasons for MANETs can be applied. In this case, the sensors are even more constrained and thus provide even less memory space than MANET devices.

Small program memory requirement: Same reasons as in the previous item.

Restrict consequences of data disclosure: The level of physical protection of sensor networks is very limited. Since sensor networks are used for military purposes, it is very important to consider the case that an adversary, the enemy, has physical access to a sensor. As argued before, we can only implement symmetric cryptographic systems in sensor networks. Hence, it is desirable to implement some additional features to prevent an attacker from compromising the entire system by compromising a single sensor. As in MANETs, the possibility of a detection of a compromised sensor should be considered.

Scalability: Taking the large number of nodes into account, solutions for sensor networks need to scale well with the number of sensors in the network.

Fault tolerance: Sensors are very prone to failure. For this reason, sensor networks consist of a large number of sensors to gain redundancy. Therefore, protocols implemented in sensor networks should be fault tolerant.

4.3 Protocol Stages

The desired protocol should authenticate mobile devices and securely establish a session key among them. After or during a successful authentication, a session key should be established for encrypting all further communications among the devices. Once an authentic key data, i.e. either a secret key or an authentic copy of a public key, is shared among the devices, the same key data will be used for all following authentications. A new encryption key will be established for each session. The use of different keys for authentication and encryption is desirable for many reasons [40]. The protocol can be executed in the *running system*, i.e. after the network initialization phase (if there was any). The first type of protocol we introduce consists of three stages, the second of two.

4.3.1 3-Stage Protocol

1. **Pre-Authentication**

   The first stage is the pre-authentication phase of the devices that authentically exchange data. In symmetric schemes, the secret keys are exchanged in this phase. In asymmetric schemes the long-term public keys and optionally other public data are authentically exchanged. This phase is only performed once among the same set of devices. The process needs only to be repeated if the keys are revoked or expired. The next time the same devices wish to establish a secure channel, they can skip this stage and directly start with the authentication.

2. **Authentication**

   In the second stage, the authentication stage, the participants start the authentication protocol. If authentication of one device fails, the protocol stops and further countermeasures might be taken, such as revoking the key of the rejected device.

3. **Session Key Establishment**

   After the successful protocol execution, the devices start to establish a session key in the third protocol stage. It has to be kept in mind that all session keys need to be established over an authentic channel. Otherwise, Oscar could overtake Alice's role after her successful authentication to Bob. To overcome this attack, either authentic keys need to be used for the message exchange during the session key establishment, or the protocol steps need to be combined with the authentication protocol. In the latter case, the exchanged messages of the authentication protocol would already contain the information that is needed to derive the session key.

If only one message needs to be authentically exchanged between the two talking nodes and the same nodes will probably not talk to each other again, stage 3 becomes redundant. The few messages to be authentically exchanged can either be included in the authentication protocol, or can be encrypted using the authenticated key material.

4.3.2 2-Stage Protocol

We introduce a second type of protocol which is suitable for all applications that do not provide a secure channel for the pre-authentication. Since we cannot exchange any data prior to the execution of the authentication protocol, a pre-authentication phase cannot be provided.

As discussed in Section 2.4 protocols in the ID-based model and the self-certified public key model do not require a pre-authentication phase. In the first model, commonly known information is used as both public key and certificate at the same time, and in the second model, the certificate is embedded in the public key itself. Thus, key data does not need to be exchanged prior to the protocol execution. The communicating devices only need to be in possession of some system parameters. Since we do not need a pre-authentication phase, the protocol consists of two stages only. These two phases are the two latter phases of the 3-stage protocol, namely the authentication and the session key establishment.

5 Summary and Future Prospects

We conclude that some commercial ad hoc network applications can be securely and efficiently implemented by symmetric solutions. The Bluetooth model is applicable for all PANs, in which a user can set up all of his/her devices with one password, or an administrator is able to set up all authorized devices in order to share network resources. The resurrecting duckling model is suitable for all applications where people or devices, who already trust each other, are located in a small area.

An asymmetric approach which seems to be suitable for mobile device-terminal connections is the exchange of public keys over a location-limited channel, as introduced in [4]. This approach could be implemented in some civil applications, such as virtual classrooms, internet access points, and all communications between PDAs and laptops of different users, who meet and would like to securely exchange data. This approach is also limited to networks with a small number of devices that provide moderate computational power. All approaches in the distributed CA or self-organization model are only suitable for networks with a large number of nodes. MANETs are not guaranteed to consist of a certain number of nodes all the time, especially at the time of the system's set up. In addition, we believe that all approaches using threshold schemes are not efficient in terms of the computational and communication overhead.

We believe that a general trend in embedded processor technology is that processors are becoming more and more powerful, which enables constrained devices, e.g. MANET devices, to perform complex computations such as modular exponentiations. Consequently, applicable protocols could require modular exponentiations or similar heavy computations, as needed in asymmetric schemes, for instance. On the other hand, we believe that the battery power of devices will remain limited which requires the number of computations to remain limited. Another trend we predict is that internet will become accessible almost everywhere. This would enable access to a backbone everywhere.

We plan to further analyze authentication and key exchange protocols that were proposed for an implementation in ad hoc networks. We will use the design goals that we have derived in this report to verify if the proposed protocols are applicable in ad hoc networks. We plan to develop some protocols in the different authentication models that require only cheap and balanced computations. If internet access can be provided to the network, solutions that require a CA or proxies would become applicable in many scenarios. In particular, the exploration of proxies used in ad hoc network application will be one of our next investigations. Using proxies would enable us to shift heavy computations from the constrained devices to the more powerful trusted proxy servers.

References

[1] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, ISBN 0471389226, 2001.

[2] W.A. Arbaugh, N. Shankar, and Y.C. J. Wan. Your 802.11 Wireless Network has No Clothes, available at `http://www.cs.umd.edu/~waa/wireless.pdf`

[3] N. Asokan and P. Ginzboorg. Key Agreement in ad hoc networks, *Computer Communications*, vol. 23, no. 17, 2000, pp. 1627-1637.

[4] D. Balfanz, D.K. Smetters, P. Stewart, and H. Chi Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks, *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS '02)*, 2002.

[5] S.M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, IEEE Computer Society, ISBN: 0-8186-2825-1, 1992, pp. 72-84.

[6] F. Bennett, D. Clarke, J.B. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded Mobile Networking, *IEEE Personal Communications*, vol. 4, no. 5, 1997, pp. 8-15.

[7] Bluetooth SIG, *Specification of the Bluetooth system*, Version 1.1; February 22, 2001, available at `https://www.bluetooth.com`

[8] D. Boneh, G. Durfee, and M. Franklin. Lower bounds for multicast message authentication. *Advances in Cryptology- EUROCRYPT '2001*, B. Pfitzmann (Ed.), Springer-Verlag, LNCS 2045, 2001, pp. 434-450.

[9] M. Bohge and W. Trappe. An authentication framework for hierarchical ad hoc sensor networks, *Proceedings of the 2003 ACM workshop on Wireless security*, ISBN:1-58113-769-9, ACM Press, 2003, pp.79-87.

[10] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11, available at `http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html`, 2001.

[11] L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks, *Mobile Network Applications*, special issue on Mobile Ad Hoc Networks, Kluwer Academic Publishers, 2003, vol. 8, no. 5, pp. 579-592.

[12] L. Buttyán and J.-P. Hubaux. Report on a Working Session on Security in Wireless Ad Hoc Networks, *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM Press, 2003, vol. 7, no. 1, pp. 74-94.

[13] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Self-Organized Public-Key Management for Mobile Ad Hoc networks, *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, 2003, pp. 52-64.

[14] W.B. Mueller and W. Nobauer. Cryptanalysis of the Dickson-scheme, *Advances in Cryptology-EUROCRYPT '85*, LNCS 219, Springer-Verlag, 1986, pp. 50-61.

[15] S. Fluhrer and S. Lucks. Analysis of the E0 Encryption System, *8th Annual International Workshop on Selected Areas in Cryptography (SAC 2001)*, S. Vaudenay, A.M. Youssef (Eds.), LNCS 2259, Springer-Verlag, 2001, pp. 2267-287.

[16] C. Gentry. Certificate-Based Encryption and the Certificate Revocation Problem, *Advances in Cryptology- EUROCRYPT '2003*, E. Biham (Ed.), LNCS 2656, Springer-Verlag, 2003, pp. 272-293.

[17] M. Girault. Self-certified public keys, *Advances in Cryptology- EUROCRYPT '91*, D.W. Davies (Ed.), LNCS 547, Springer-Verlag, 1991, pp. 490-497.

[18] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks, *9th ACM conference on Computer and Communications Security*, ISBN:1-58113-612-9, ACM Press, 2002, pp. 41-47.

[19] S. Gokhale and P. Dasgupta. Distributed Authentication for Peer-to-Peer Networks, *Symposium on Applications and the Internet Workshops 2003 (SAINT'03 Workshops)*, IEEE Computer Society 2003, ISBN 0-7695-1873-7, 2003, pp. 347-353.

[20] G. Gong and L. Harn. Efficient Lucas-type public key cryptosystems, *Proceedings of 1996 International Conference on Cryptology and Information Security*, 1996.

[21] G. Gong and L. Harn. Public-key cryptosystems based on cubic finite field extensions, *IEEE Trans. on Inform. Theory*, 1999, vol. 45, no. 7, pp. 2601-2605.

[22] G. Gong, L. Harn, and H. Wu. The GH Public-Key Cryptosystem, *Proceedings of Selected Areas in Cryptography (SAC) 2001*, LNCS 2259, Springer-Verlag, 2001, pp. 284-300.

[23] Official HomeRF Homepage, `http://www.homerf.org`

[24] IEEE 802.11, Standard Specifications for Wireless Local Area Networks, `http://standards.ieee.org/wireless/`

[25] IEEE 1363, Standard Specifications for Public-Key Cryptography, 2000. `http://grouper.ieee.org/groups/1363/index.html`

[26] Offical Infrared Data Association (IrDa) Homepage, `http://www.irda.org`

[27] HP's cooltown project, `http://cooltown.hp.com/cooltownhome/index.asp`

[28] J.-P. Hubaux, L. Buttyán, and S. Čapkun. The Quest for Security in Mobile Ad Hoc Networks, *ACM Symposium on Mobile Ad Hoc and Computing –MobiHOC 2001*, 2001, pp. 146-155.

[29] M. Jacobsson and S. Wetzel. Security weaknesses in Bluetooth, *Cryptographer's Track at RSA Conference 2001*, D. Naccache (Ed.), LNCS 2020, Springer-Verlag, 2001, pp. 176-191.

[30] D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, vol. 353, Kluwer Academic Publishers, 1996.

[31] J. Katz, R. Ostrovsky, and M. Yung. Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords, *Advances in Cryptology- EUROCRYPT '2001*, B. Pfitzmann (Ed.), LNCS 2045, Springer-Verlag, 2001, pp. 475-494.

[32] A. Khalili, J. Katz, and W. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks, *2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*, IEEE Computer Society, ISBN 0-7695-1873-7, 2003, pp. 342-346.

[33] O. Kömmerling and M.G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors, *1st USENIX Workshop on Smartcard Technology*, 1999, pp. 9-20.

[34] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks, *International Conference on Network Protocols (ICNP) 2001*, 2001.

[35] L. Lamport. Password authentication with insecure communication, *Communication of the ACM*, vol. 24, no. 11, 1981, pp. 770-772.

[36] A.K. Lenstra and E.R. Verheul. The XTR public key system, *Advances in Cryptology-CRYPTO '2000*, LNCS 1880, Springer-Verlag, 2000, pp. 1-9.

[37] D. Liu and P. Ning. Effcient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks, *Proceedings Network and Distributed System Security Symposium Conference (NDSS) '03*, 2003.

[38] D. Liu and P. Ning. Location-Based Pairwise Key Establishments for Static Sensor Networks, *1st ACM Workshop Security of Ad Hoc and Sensor Networks (SASN) '03*, ISBN:1-58113-783-4, ACM Press, 2003, pp. 72-82.

[39] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-Securing Ad Hoc Wireless Networks, *Seventh IEEE Symposium on Computers and Communications (ISCC '02)*, 2002.

[40] A.J. Menezes, P.C. von Orschot, and S.A. Vanstone. *Handbook of Applied Cryptography*, 1997 by CRC press LLC.

[41] T.S. Messerges, J. Cukier, T.A.M. Kevenaar, L. Puhl, R. Struik, and E. Callaway. A security design for a general purpose, self-organizing, multihop ad hoc wireless network, *1st ACM workshop on Security of ad hoc and sensor networks (SASN) '03*, ISBN:1-58113-783-4, ACM Press, 2003, pp. 1-11.

[42] National Institute of Standards and Technology NIST, Wireless Ad Hoc Network Projects, `http://w3.antd.nist.gov/wahn_home.shtml`

[43] C. Perkins. Ad Hoc On Demand Distance Vector (AODV) Routing, Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997, 1997.

[44] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *IEEE Symposium on Security and Privacy*, 2000, pp. 56-73.

[45] A. Perrig, R. Szewcyk, V. Wen, D. Culler, and J.D. Tygar. SPINS: Security Protocols for Sensor Networks, *Mobile Computing and Networking*, 2001, pp. 189-199.

[46] A. Perrig, R. Canetti, D. Song, and J.D. Tygar. Efficient and Secure Source Authentication for Multicast, *Network and Distributed System Security Symposium '01 (NDSS '01)*, 2001.

[47] A.O. Salako. Authentication in Ad hoc Networking, *In Proceedings of London Communications Symposium 2002*, 2002.

[48] N.B. Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks, *In Proceedings of the* 4^{th} *ACM/SIGMOBILE MobiHoc*, ISBN 1-58113-684-6, 2003, pp. 13-24.

[49] A. Shamir. Identity-based Cryptosystems and Signature Schemes, *Advances in Cryptology-CRYPTO '84*, G.R. Blakley, D. Chaum (Eds.), LNCS 196, Springer-Verlag, pp. 47-53, 1984.

[50] P. Smith. LUC public-key encryption, *Dr. Dobb's Journal*, 1993, vol. 18, no. 1, pp. 44-49.

[51] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks, *In Proceedings of the 7th International Workshop on Security Protocols*, B. Christianson, B. Crispo, J.A. Malcolm, and M. Roe (Eds.), LNCS 1796, Springer-Verlag, pp. 172-194, 1999.

[52] F. Stajano. The Resurrecting Duckling - what next?, *Proceedings of the 8th International Workshop on Security Protocols*, B. Christianson, B. Crispo, and M. Roe (Eds.), LNCS 2133, Springer-Verlag, pp. 204-214, 2000.

[53] F. Stajano. *Security for Ubiquitous Computing*, John Wiley & Sons, ISBN 0470844930, 2002.

[54] J. Walker. Unsafe at any key size; An analysis of the WEP encapsulation, *Tech. Rep. 03628E, IEEE 802.11 committee*, available at `http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip`, 2000.

[55] R. Want, A. Hopper, V. Falcão, J. Gibbons. *The Active Badge Location System*, Olivetti Research Ltd., 1992.

[56] A. Weimerskirch and G. Thonet. A Distributed Light-Weight Authentication Model for Ad-hoc Networks, *In Proceedings of the 4th International Conference on Information Security and Cryptology (ICISC 2001)*, LNCS 2288, 2002, pp. 341-354.

[57] A. Weimerskirch and D. Westhoff. Zero Common-Knowledge Authentication for Pervasive Networks, *Tenth Annual International Workshop on Selected Areas in Cryptography (SAC 2003)*, 2003.

[58] A. Weimerskirch and D. Westhoff. Identity Certified Authentication for Ad-hoc Networks, *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (SASN)*, 2003, ACM Press, ISBN:1-58113-783-4, 2003, pp. 33-40.

[59] L. Zhou and Z.J. Haas. Securing Ad Hoc Networks, *IEEE Network Journal*, vol. 13, no. 6, 1999, pp. 24-30.