Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation

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

Abstract

In this work, we introduce two full functional identity-based authentication and key exchange (IDAKE) schemes for mobile ad hoc networks (MANETs). Therefore, we utilize some special features of identity-based cryptographic (IBC) schemes, such as pre-shared secret keys from pairings and efficient key management, to design MANET-IDAKE schemes that meet the special constraints and requirements of MANETs. As part of these schemes, we present the first key revocation and key renewing algorithms for IBC schemes. The former algorithm uses a new concept of neighborhood watch. We introduce a *basic MANET-IDAKE* scheme in which a trusted third party (TTP) initializes all devices before they join the network and a *fully self-organized MANET-IDAKE* scheme that does not require any central TTP. The schemes bootstrap the security in MANETs and enable the use of authentication, key exchange, and other security protocols in a variety of applications. Furthermore, we present an extremely efficient yet secure IDAKE protocol that can be used in the presented schemes. Finally, we provide a security and performance discussion of the presented MANET-IDAKE schemes and IDAKE protocol.

1 Introduction

1.1 Motivation

The number of applications that involve wireless communications among mobile devices is rapidly growing. Many of these applications require wireless networks to be spontaneously formed by the participating mobile devices themselves. Such networks are referred to as *mobile ad hoc networks (MANETs)*. The idea behind MANETs is to enable connectivity among any arbitrary group of mobile devices everywhere, at any time. Slowly people realize that security is of paramount importance in MANETs. However, the special properties of ad hoc networks, such as the lack of infrastructure, absence of trusted third parties (TTPs), as well as the constraints of the devices and the communication channel, make implementing security a very challenging task. Among the major challenges are: bootstrapping security, providing authentication and key exchange, and enabling key revocation and key renewing in public key infrastructures (PKIs). Prior to the execution of authentication and other security protocols, all nodes need to share some authentic credentials to be able to prove their identity to each other. We refer to the initial exchange of credentials as *pre-authentication*. Pre-authentication requires some kind of secure channel which is very difficult to achieve in most ad hoc network applications. Many schemes to solve the pre-authentication problem and provide authentication and key establishment in ad hoc networks have been proposed [1, 2, 6, 9, 13, 14, 18–20, 22, 28, 29]. However, these schemes are only suitable for certain ad hoc network applications due to their restrictive requirements or demanding computational and communications costs. Furthermore, many of the proposed PKI and all IBC schemes proposed for MANETs do not provide algorithms for certificate or key revocation, respectively, and key renewing. We discuss the previously proposed schemes and their limitations in Section 2.

Due to the limitations and shortcomings of proposed scheme, we believe that there is a need to continue developing authentication and key exchange schemes for MANETs, including algorithms for key revocation and renewing. Until now, the role of IBC schemes as enabler for security in ad hoc networks has not been thoroughly explored. We believe that IBC schemes have some distinctive features that make them an excellent tool to bootstrap security in some ad hoc network applications.

1.2 Contributions

The contributions of this paper are three-fold:

1. We identify some distinctive features of IBC schemes that qualify those schemes as an alternative public key scheme to secure ad hoc networks.

2. We introduce two full functional identity-based authentication and key exchange (IDAKE) schemes for MANETs. In particular, we present a *basic MANET-IDAKE* scheme in which a key generation center (KGC) initializes all devices before they join the network and a *fully self-organized MANET-IDAKE* scheme without any central KGC, where all tasks are performed by the network nodes themselves. We are the first to introduce key revocation and key renewing mechanisms for IBC schemes. The schemes are designed to address the special constraints, requirements, and diversity of MANETs. For instance, the schemes only use symmetric cryptography and pairing-based keys which makes them very efficient.

3. We present a lightweight identity-based authentication and key exchange protocol for MANETs that can be implemented in the presented MANET-IDAKE schemes. The protocol is extremely efficient and can be proven secure without perfect forward secrecy in the Canetti-Krawczyk model [10].

1.3 Outline

The remainder of the paper is organized as follows. In the next section, we summarize some previous work on authentication and key exchange schemes for ad hoc networks, including key revocation. Next, in Section 3, we briefly review IBC schemes and discuss some of their distinctive features that are attractive for ad hoc networks. In Section 4, we introduce two full functional MANET-IDAKE schemes including a lightweight IDAKE protocol. We analyze the security and performance of the presented IDAKE schemes and protocol in Section 5. Finally, we draw some conclusions in the last section.

2 Previous Work

In this section we discuss some of the proposed authentication and key exchange schemes for ad hoc networks, including key revocation and renewing. Due to their excellent performance, symmetric crypto schemes seem well-suited for MANETs. However, key distribution in such schemes is a major problem. Due to the absence of an on-line key distribution center in MANETs, pre-authentication in symmetric schemes requires either the proximity of the communicating devices [28], or some other kind of out-of-band channel [1, 6, 19]. These limitations and the need for non-reputable communications in some applications triggered the research on PKI and IBC schemes for MANETs.

2.1 Public Key Solutions

An authentic channel is needed to exchange public keys in PKIs during pre-authentication. This channel is established by the use of either location limited channels or public key certificates. In the first approach, public keys are directly exchanged and certificates are redundant. This solution requires the close proximity of the users or the knowledge of the geographical location of the communication partner [2, 9]. In the second approach, certificates are used to bind public keys to an identity. Therefore, a certification authority (CA) is needed to issue, distribute, revoke, and renew public key certificates. Due to the absence of a central CA in MANETs, many proposed solutions emulate a CA in the network. We refer to such a CA as *internal CA*. Internal CAs can be implemented by (k, n)-threshold schemes to distribute the power and tasks of the CA to a group of special nodes [29] or all network nodes [22]. The former solution suggests collaboratively revoking keys but no algorithm is introduced. In fact, a revocation scheme in this solution would require threshold signatures which is not defined in [29]. Note that protocols utilizing threshold schemes are very demanding in terms of computational and communication costs. In another approach, every network node acts as an internal CA, i.e. nodes issue and distribute their own public keys and sign others in a PGP manner [18]. This approach is based on two assumptions: (1) trust is transitive and (2) a trusted path consisting of signed public keys (certificates) exists between any pair of communicating nodes. The performance of this scheme highly depends on the length of the trusted path and is generally hard to predict. An algorithm for key revocation is not described in [18].

In another class of certificate-based PKI schemes for MANETs, network devices are initialized by an external off-line KGC. Proposed revocation schemes for PKIs without internal CA introduce so-called 'accusation' schemes [12, 22]. Here, each node can accuse other nodes to be malicious or compromised. If the number of accusations is greater than a certain threshold δ, the certificate is considered to be revoked. All accusations need to be frequently broadcasted in order to inform all nodes about recent revocations and changes. For instance, [22] proposes a sign&broadcast approach to securely distribute the accusation tables. However, the event of a newly joining node in this scheme would require signing each individual accusation and storing these signatures together with the accusations. Newly joining nodes would then receive signed accusation tables from its neighbors and would need to verify the signed table and each accusation in the table. In a network with N nodes the maximum number of required verifications is N^2, or $\delta N + 1$ if we assume that, once revoked, user do not accuse a node any longer. This is clearly too demanding, not only in MANETs. On the other hand, [12] does not secure the propagation of accusation tables. Instead, newly joining nodes derives their own accusation tables by evaluating the tables of all other network nodes. This requires frequent broadcasts of all tables which adds a lot of communication load to the network.

2.2 Identity-Based Solutions

Recently, identity-based cryptographic (IBC) schemes have been considered for securing ad hoc networks [13, 20]. Both papers suggest emulating an internal KGC using (k, n)-threshold schemes, as previously introduced for internal CAs in PKIs. The key management in both solutions is entirely self-organized by the network nodes and the authors claim that their schemes are more efficient than fully self-organized PKIs because of the efficient key management of the underlying IBC schemes. However, using threshold schemes also introduces a lot of computational and communication overhead to IBC schemes. Furthermore, no protocols for key establishment and authentication are proposed for the IBC schemes. In [13],the authors suggest using a pre-shared key for encryption. However, static keys should never be used for encryption and key establishment protocols are desirable. Both schemes do not introduce key revocation and key renewing algorithms for their schemes. As far as we know, no key revocation or renewing algorithm for IBC schemes has been introduced yet, neither for general networks nor for MANETs.

3 Identity-Based Cryptographic Schemes in MANETs

3.1 Preliminaries

In 1984, Shamir introduced the first IBC scheme [27]. However, Shamir's scheme can only be used as signature scheme. His quest for an ID-based encryption scheme remained unanswered until 2001, when Boneh and Franklin introduced the first ID-based encryption scheme from the Weil pairing [7]. Much research on ID-based schemes from Weil and other bilinear pairings has been carried out ever since, including encryption, signature and authentication schemes, e.g. [7, 15, 24], respectively. For the remainder of the paper we will limit our focus on pairing-based IBC schemes, which we refer to as *BF schemes*.

The main feature of IBC schemes is the use of self-authenticating public keys. Since identities are used as public keys, user identities and their corresponding public keys do not need to be bound by certificates or any other means. Because the public keys Q_i are predetermined in IBC schemes, the private keys d_i are derived from the corresponding public keys. For that reason, IBC schemes require a KGC to generate and distribute the private keys during the initialization of all network nodes. The KGC delivers the private keys d_i over a secure channel. Consequently, the KGC is a key escrow in all ID-based schemes. To limit the validity period of an IBC-based public key, an expiry date can be easily embedded in the key itself, e.g. by concatenating an expiry date to the public key [7]. The public key of a node ID_i could then look like in Equation 1 below. Only if node ID_i is in possession of the matching private key d_i that corresponds to the date, he can sign or decrypt messages.

$$Q_i = H_1(ID_i || \text{`expiry date'}) \tag{1}$$

In an IBC scheme, every node in the network is able to derive the public key Q_i of a communication partner ID_i in the network without the need to exchange any data. In addition to pre-shared public keys, all pairs of nodes ID_i and ID_j in a pairing-based IBC scheme are able to compute a pairwise pre-shared secret key K_{ij} in a non-interactive fashion [26]. The pre-shared secret keys are computed according to Equation 2 and have been used in authenticated encryption schemes [24] and authenticated key agreement protocols [8].

$$K_{ij} \quad = \quad \hat{e}(d_i, Q_j) = \hat{e}(Q_i, d_j) \tag{2}$$

For the key computation both parties compute the bilinear mapping $\hat{e}(\cdot)$ over their own private key d_{ID} and the public key Q_{ID} of the desired communication partner. Note that the KGC is able to compute all pre-shared keys.

3.2 Distinctive Features of IBC Schemes in MANETs

We believe that IBC schemes are an attractive security solution for many MANET applications and we discuss some special features of IBC schemes in the following:

A. IBC schemes provide *implicit* and *non-interactive* pre-authentication among all network nodes

B. IBC schemes provide *implicit* public key validity checks

Feature A is due to the use of identities as public keys which entails many desirable properties. IBC schemes do not require any secure channel for pre-authentication, because (1) public keys are self-authenticating and (2) known prior to communication. For the first reason no additional credentials to proof the authenticity of keys are needed in IBC schemes in contrast to public key certificates in PKIs. This offers a more intuitive security and helps to reduce the communication overhead and required memory space. Secondly, since ID-based public keys are known in advance, the bandwidth requirements can be further reduced because public keys do not need to be exchanged.

Feature B allows an easy way to check whether a public key is valid. Note that we refer to *valid public key* when keys are not expired. However, validity does not indicate whether a key is revoked. As described in the previous section the expiry date can be directly embedded in the public keys themselves. When verifying a signature in an ID-based signature scheme we check the validity of the keys at the same time, and in case of an ID-based encryption scheme, only users with valid keys are able to decrypt. Contrarily, in PKI schemes public keys have their expiry date listed in a public key certificate. When nodes receive public keys and the corresponding certificates they explicitly need to check the date in the certificate to see whether the key is expired or not.

All BF-schemes offer an additional attractive property for MANETs:

C. Paring-based IBC schemes provide a pairwise secret key K_{ij} (Equation 2) that is pre-shared in a *non-interactive* fashion

The additional Feature C of paring-based schemes gives us all the benefits of symmetric key schemes without the need of a secure channel during pre-authentication. Each pair of nodes ID_i and ID_j in the network shares a secret K_{ij}, before ever having communicated with each other. This feature makes protocol messages during pre-authentication redundant and thus saves bandwidth. The pre-shared keys can be used to enable mutual authentication, key exchange, secure routing, and more at low computational and communication costs. Note that pairwise secret keys can be derived in all PKIs too, e.g. static Diffie-Hellman keys. However, those keys require the authentic exchange of public keys and are thus not derived in a non-interactive fashion.

After discussing the benefits of IBC schemes we now briefly comment on some known drawbacks of these schemes. Generally, the special role of the KGC as a key escrow is considered as a disadvantage. Usually threshold schemes for distributed KGCs are introduced for this matter [7]. However, we believe that there are more efficient and convenient solutions for MANETs. For instance, curious-but-honest KGCs can be prevented from eavesdropping by executing a Diffie-Hellman (DH)-like key agreement protocol [11]. Active attacks of dishonest KGCs can not be fully prevented, however, similar attacks can be executed by dishonest CAs in PKIs. The likelihood of successful attacks by malicious KGCs or CAs in MANETs has been shown to be significantly lower than in other conventual networks due to the short communication range and mobility of the users [16]. Another drawback of IBC schemes is the requirement of a confidential and authentic channel between the KGC and each network node for the secure distribution of the private keys. However, when using a blinding technique as proposed in [21] an authentic channel is sufficient. Finally, providing key revocation in IBC schemes is considered as a problem. We would like to point out that providing key revocation in IBC schemes is as crucial as in PKIs and in any case challenging to implement in MANETs. We introduce a revocation scheme for IBC schemes in MANETs in the next section.

4 MANET-IDAKE Schemes

We now present a description of two fully functional identity-based authentication and key exchange (IDAKE) schemes for MANETs.

4.1 Choosing Identities

Before introducing the algorithms of the actual MANET-IDAKE schemes, we need to discuss how the identities of all network nodes could be chosen. Identities must be *unique* for each entity in the network. Furthermore, identities must be *unchangeably bound* to an entity for its entire *lifetime* and the identity is *not transferable*. The string of information that can be used as identity depends on the application. We need to consider who needs to be authenticated or identified in the network, and thus the identities have to be chosen according to the requirements of particular applications. Generally, we can distinguish three cases of entities an identity is bound to: (1) a user operating a network node, i.e. the ID string corresponds to the user, e.g. the user's email address; or (2) a device, i.e. the ID is bound to the hardware, e.g. the MAC address; or (3) a network interface, in that case the ID might be derived from the IP address.

For example, if an application enables two users to securely communicate with each other, the use of user-dependent IDs seems desirable. Note that in that case multiple users are able to share the same device. In sensor networks and other ad hoc networks in which user do not operate the devices, the MAC address seems to be a good choice, since the address satisfies all required properties of suitable identities. A combination of both previous approaches is also thinkable using two different sets of ID-based keys to meet the requirements of different protocol layers. The third scenario might be of interest in some special applications. However, it is not feasible in many MANETs because network addresses such as IP addresses might be dynamic or do not exist at all.

4.2 Key Renewal

Every key management scheme, both IBC and PKI, should provide a key renewal algorithm to enable nodes to obtain a new key after key expiration or compromise. In MANETs without internal KGC, key renewal is an off-line operation that requires nodes to request new keys from an external KGC. However, this might not be possible in some applications and thus key renewing cannot be provided. Hence, once a key is compromised or expired, the node is excluded from the network. Note that the limitations are the same in symmetric schemes and PKIs without internal TTP. In schemes with internal KGC, nodes authenticate themselves to the KGC, i.e. to k nodes in a threshold scheme, to request a new key. Schemes that allow key renewal without such an authentication before the expiry date, e.g. [23], do not prevent adversaries to renew compromised keys. Here, short periods of key usage without new authentications only reduce the amount of available ciphertext encrypted under the same key and thus make cryptanalysis harder.

The frequency of required key renewals is a system parameter that needs to be chosen according to the application. The more often keys are renewed the less likely is key compromise. Hence, such schemes provide a higher security level, and in addition, key revocation becomes redundant [23]. Frequent key renewals put the computational and communication burden on the nodes that wish to communicate, instead on the entire network, as it is the case for internal revocation schemes. However, it needs to be further studied for individual applications whether frequent renewals or revocation schemes are more efficient. This is out of the scope of this paper.

As described earlier, the expiry date is directly embedded in ID-based public keys, see Equation 1. However, this key format is only sufficient in schemes without revocation. In schemes with immediate key revocation, nodes need to be able to request key renewal even before the expiry date, e.g. in case of key compromise. For instance, node ID_i requests a new key pair at $date_1$ after its old keys with expiry date $date_2$ were compromised, with $date1 < date_2$. Since the identity ID_i is unchangeable in IBC schemes, the key cannot be issued for the same expiry date because it would result into the old, compromised, keys. However, issuing the new keys with a new expiry $date_3 > date_2$ might not be feasible either, because a node ID_i might only be eligible to possess keys until $date_2$. Furthermore, in IBC schemes expiry dates need to be chosen in a predictable manner, e.g. fixed intervals, such that nodes do not need to exchange public keys. Otherwise one of the features of IBC schemes would be lost. Hence, we need to add some additional data to the public key that can be changed with every key renewal. We use the following format as given in Equation 3 below.

$$Q_i = H_1(ID_i || \text{`expiry date'} || \text{`version \#'}) \tag{3}$$

For instance, upon compromise of $Q_i = H_1(ID_i||date_1||v)$, node ID_i can request a new key Q_i' before $date_1$, with $Q_i' = H_1(ID_i||date_1||v + 1)$. Note that if the date of request is close to $date_1$, the KGC can issue two key pairs, one with the expiry date $date_1$ and version number $v + 1$ and another one with date $date_2$, with $date_2 > date_1$, and version number 1. Node ID_i can then switch to the second key once the first is expired and does not need to request a new key pair again. Note that the version number always starts with 1 for every new expiry date and is incremented with each key renewal for the same date.

A KGC (external or internal) can usually not distinguish between malicious nodes whose keys have been revoked because of bad behavior or honest nodes that have been compromised. Therefore, malicious nodes could always request new keys once their malicious behavior is detected and their keys have been revoked. Note that malicious nodes are acting under their true identities and thus can successfully authenticate themselves to the KGC. To restrict the power of such malicious nodes, we choose a maximum version number v_{max}, i.e. the number of key renewals for the same expiry date is restricted. Clearly, a node that requests more than v_{max} key renewals is either malicious or not able to appropriately protect its key data. Parameter v_{max} depends on the length of expiry intervals, generally the longer the intervals the greater v_{max} should be chosen. However, in general v_{max} should be rather small.

4.3 Key Revocation

If frequent key renewals cannot be provided, every node in a MANET needs to be able to verify whether a public key is revoked. Public key revocations need to be handled within the network, because nodes need to be able to immediately verify the status of a public key. So far in all IBC schemes, i.e. general schemes and schemes especially designed for MANETs, revocation referred to embedding an expiry date in the public key. We would like to stress that this is not sufficient because nodes need to be able to revoke keys before they expire, e.g. in the case of key compromise or malicious behavior. In order to provide key revocation in MANETs we need four mechanisms. First, nodes need to be able to revoke their own public key, which we refer to as *harakiri*. Second, nodes can revoke the public keys of compromised or suspicious nodes, which we refer to as *accusation*. Third, we need a mechanism to inform all nodes in the network about these revocations. And last, we need a mechanism for newly joining to obtain a list of all revoked or accused public keys.

Neighborhood Watch. We now introduce a new accusation scheme that enables nodes to check whether keys are compromised in static ad hoc networks. We call the scheme *neighborhood watch* and use it as a basic building block in the revocation scheme proposed in the next section. Neighborhood watch provides mechanisms for harakiri messages, accusations, and newly joining nodes, but does not provide a mechanism to inform other nodes about accusations. Hence, the scheme is not a revocation scheme and rather a local compromise check. We believe that any node ID_i can effectively monitor its one-hop neighbors, denoted as neighborhood $\mathcal{N}_i$. Therefore, we assume that all nodes know the identities of their one-hop neighbors. In most cases the information is already known because it is needed in most routing protocols. If not provided by routing or other lower layers, nodes need to explore their neighborhood by sending *hello* messages that contain their identities and listening to the responses. Nodes monitor their neighborhood for suspicious behavior, such as frequent packet drops and an unusually large number of sent messages. When such a behavior is observed, the respective node is marked as suspicious. In our scheme every node ID_i observes nodes $t_1, \ldots, t_{e_i}$ with $t_j \in \mathcal{N}_i$ and $j \in \{1, \ldots, e_i\}$, where $e_i = |N_i|$ is the number of one-hop neighbors. Note that t_j is used instead of identity ID_i, with $t_j = ID_i$, to enable the numeration of the nodes from 1 to e_i, because identities ID_i do not have a specific order. This notation makes the description of the scheme easier, however, in an implementation the first column would contain the identities of the nodes. After a monitoring time T_m, node ID_i creates and stores an accusation matrix

$$\mathcal{AL}_i = \begin{pmatrix} t_1 & (date_1, v_1) & c_1 \\ \vdots & \vdots & \vdots \\ t_{e_i} & (date_{e_i}, v_{e_i}) & c_{e_i} \end{pmatrix}$$

The flag $c_j \in \{0, 1\}$ indicates the status of Q_j. In particular, $c_j = 1$ if t_j is suspicious or Q_j

expired or revoked by harakiri, otherwise $c_j = 0$. $\mathcal{AL}_i$ is updated every time ID_i observes suspicious behavior of one of its neighbors. Once the flag c_j is set, it will not be reset to zero until a new key is received for t_j. Formerly accused or expired nodes t_j broadcasts their new keys (new expiry date or version number) to $\mathcal{N}_j$ and all recipients update the j-th row in their $\mathcal{AL}$ accordingly. A newly joining node ID_n sends its identity to all one-hop neighbors. After a period T_m, ID_n creates its own accusation matrix $\mathcal{AL}_n$ and the neighbors t_j create an entry in their accusation matrices $\mathcal{AL}_j$.

If a node ID_i realizes that its keys have been compromised, it sends a harakiri message hm_i consisting of its public key, ID, and a revocation message to all $t_j \in \mathcal{N}_i$. To prevent malicious harakiri messages, hm_i is authenticated using a secure MAC function $f(\cdot)$ with the pre-shared keys from Equation 2, i.e. $hm_i = (f_{K_{ij}}(Q_i, ID_i, \text{'revoke'}), Q_i, ID_i, \text{'revoke'}),\ \forall\ t_j \in \mathcal{N}_i$. Upon receiving hm_i, all nodes t_j verify the authenticity of the message by using their pre-shared keys K_{ij}. If hm_i is successfully authenticated, nodes t_j consider Q_i to be revoked and set $c_k = 1$, with $t_k = ID_i$, in their accusation matrices.

In the proposed scheme a node ID_i only forwards messages to/from a node t_j in $\mathcal{N}_i$ if $c_j = 0$ in $\mathcal{AL}_i$. This mechanism secures multi-hop communications in a way that each node ensures the trustworthiness of the next hop. For instance, if a node ID_s wants to talk to a node ID_r that is more than one hop away, i.e. $ID_r \notin \mathcal{N}_s$, ID_s finds a routing path such that the first hop t_j in $\mathcal{N}_s$ is trustworthy, i.e. $c_j = 0$. For the next hop, t_j chooses a trustworthy node t_k in $\mathcal{N}_j$ with $c_k = 0$. This procedure of forwarding the message to a trustworthy neighbor is repeated until the packet reaches the destination ID_r. In that way a path of trust is generated. If the last node ID_l in the routing path delivers the message to ID_r, we can assume that ID_r is not suspicious and thus its public key is believed to be uncompromised. In the case of a routing request to a locally suspicious node ID_r, ID_l drops the packet and sends a notification message to the sender ID_s. The notification message is secured by a chain of MACs with the pre-shared key of the last node (K_{ls}) and all intermediate nodes ID_i (K_{is}) as input.

4.4 Basic MANET-IDAKE Scheme

We now introduce a MANET-IDAKE scheme with external KGC and without internal KGC. We refer to the scheme as *basic MANET-IDAKE* scheme. The scheme is specified by 6 algorithms: *(1.) Setup, (2.) Extract, (3.) Distribute, (4.) Compute Shared Key (5.) Key Renewal*, and *(6.) Key Revocation*. The scheme is based on a BF-scheme and we adopt the notions from [7]. Algorithms 1-3 and 5 are executed by an external KGC, i.e. outside the network. Particularly, Algorithm 1-3 are executed by the KGC to initialize nodes before they join the network, whereas Algorithm 5 requires current nodes in the network to contact the external KGC.

(1.) Setup [7]. On the input of a security-parameter k, the KGC selects two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, where q is a prime, and selects a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$. The map $\hat{e}$ is admissible, i.e. it is bilinear, non-degenerate, and computable as defined in [7]. The parameters are chosen in such a way that the bilinear Diffie-Hellman problem (BDH) is hard in $\mathbb{G}_1$. Furthermore, the KGC chooses a random generator $P \in \mathbb{G}_1$, picks a random number $s \in \mathbb{Z}_q^*$ and computes $P_{pub} = sP$. The parameters (s, P_{pub}) are the KGC's long-term private and public key. In addition, the KGC selects a hash function $H_1 : \{0,1\}^* \mapsto \mathbb{G}_1^*$, which is used to derive a node's public key from its identity. After the set-up is completed the KGC makes the following system parameters publicly available $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1 \rangle$. The KGC's long-term private key s is kept confidential.

(2.) Extract [7]. The KGC extracts the long-term secret key d_i for each network node with identity $ID_i \in \{0,1\}^*$. For doing so the KGC first derives the node's public key $Q_i = H_1(ID_i)$ from its identity and then computes the private key $d_i = sQ_i$.

(3.) Distribute. The KGC bootstraps all nodes with their private keys. Therefore, upon a successful authentication of node ID_i, the KGC sends the private key d_i over a secure channel to ID_i. Such a confidential and authentic channel can be established if user physically goes to the KGC or the key is embedded in the device during manufacturing. If only an authentic communication channel is available the private keys can be protected by a simple blinding technique [21].

(4.) Compute Shared Key. Whenever two nodes ID_i and ID_j wish to communicate for the first time, they each need to compute their pre-shared key K_{ij} according to Equation 2. The computation is non-interactive and no messages or keys need to be exchanged in this step. After the computation, the key is stored for future communications with the same node.

(5.) Key Renewal. A key pair (Q_i, d_i) needs to be renewed if Q_i is expired or revoked, or d_i is compromised. In any case, a node needs to be able to access the KGC for key renewal and must authenticate itself to the KGC using some credentials that identify the node. Upon successful authentication, the KGC issues a new key as described in Section 4.2. Since no internal KGC is available in the basic MANET-IDAKE, nodes need to contact an external KGC that is outside the MANET. However, frequently contacting an external KGC might be hard if not infeasible in many applications. If such a KGC access cannot be provided at any time after the network's initialization, key renewing cannot be provided.

(6.) Key Revocation. To provide key revocation in the basic MANET-IDAKE scheme we combine our neighborhood watch scheme with an accusation scheme similar to [12, 22]. Therefore we modify the accusation schemes that were proposed for PKIs in MANETs such that they can be used in IBC schemes. By doing so, the neighborhood watch scheme can be extended to work in MANETs and achieve global revocation. We achieve this by using a threshold scheme for accusations and broadcasting these accusations.

In the revocation scheme, each node ID_i maintains a public key revocation list $\mathcal{KRL}_i$ which includes the accusation list $\mathcal{AL}_i$ as defined in the neighborhood watch scheme and additionally the accusations from other nodes. The list can be represented as matrix such that

$$\mathcal{KRL} = \begin{pmatrix} t_1 & (date_1, v_1) & c_1 & a_{1,1} & \cdots & a_{1,N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_N & (date_N, v_N) & c_N & a_{N,1} & \cdots & a_{N,N} \end{pmatrix}$$

where N is the number of nodes in an m-hop neighborhood m-$\mathcal{N}$, with $m > 1$. Each row i corresponds to a node $t_i \in m$-$\mathcal{N}$. The first field contains the identity of a node $t_i = ID_j$, the second field the expiry date and version number of the current public key Q_j. The fields 4-(N+3) contain the accusation values $a_{i,1}$ - $a_{i,N}$, where value $a_{i,k} = 1$ indicates that node t_k accused t_i, and $a_{i,k} = 0$ otherwise. The third field contains a 1-bit flag c_i that, when set, indicates that Q_j is revoked. The values c_i in the $\mathcal{KRL}_s$ of node ID_s are computed as follows, $c_i = 1$ if $a_{i,s} = 1$, or $date_i$ is expired, or $a_{i,i} = 1$, or $A_i = \sum_{j=1}^{N} a_{i,j} > \delta$. Otherwise, $c_i = 0$. First condition refers to the case that t_i is marked as suspicious as part of the neighborhood watch, whereas the third case refers to key revocation of Q_i by a harakiri message. Parameter δ is the threshold for the revocation of a node, in other words accusations of at least δ different nodes are necessary to revoke a key. In that way $\delta - 1$ accusations by malicious nodes cannot revoke a key. Note that in the presented variant,

$a_{i,j}$ are either one or zero and the revocation can be checked by the sum of these values. However, it might be desirable to weigh the importance of an accusation, e.g. a node that has never been accused has more influence than a node that has been accused several times. For this purpose the scheme from [12] can be adopted to compute the weighted values $a_{i,j}$ and the sum A_i.

After describing how the $\mathcal{KRL}$ can be computed and maintained by nodes, we now need to describe how accusations can be securely propagated through the network. Every time a node ID_i observes a malicious node t_j in $\mathcal{N}_i$ it first updates its own $\mathcal{KRL}_i$ with $a_{j,i} = 1$ and $c_j = 1$, and then broadcasts an accusation message am_i to all $t_k \in \mathcal{N}_i$, with

$$am_i = (f_{K_{ik}}(ID_i, Q_j, t_j, \text{‘accuse’}), (ID_i, Q_j, t_j, \text{‘accuse’})), \forall\ t_k \in \mathcal{N}_i.$$

Note the similarities to the harakiri message in the neighborhood scheme. All recipients t_k of am_i verify the MAC and if successful, the nodes update their $\mathcal{KRL}_k$ accordingly (i.e. $a_{j,i} = 1$). Next, all t_k securely forward the message to their one-hop neighbors disjoint from $\mathcal{N}_i$ using MACs with the respective pre-shared keys. The recipients verify, update their $\mathcal{KRL}$, and forward the message. This is repeated m times. In that way, every node in an m-hop neighborhood should receive a verifiable accusation message am_i for t_j. Now, if a node ID_i wants to communicate with another node ID_j in his m-hop neighborhood m-$\mathcal{N}_i$, it first checks if Q_j is revoked, i.e. $c_j = 1$. In the case that the desired communication partner is more than m-hops away, the trusted path approach presented in the neighborhood watch scheme is applied.

We present the following harakiri mechanism for our revocation scheme, which is similar to the one of the neighborhood watch. The main difference is that here harakiri messages are sent to a m-hop neighborhood m-$\mathcal{N}_i$, instead of $\mathcal{N}_i$. Hence, all nodes in a m-hop neighborhood around ID_i know about ID_i's key compromise and will not trust the corresponding keys any longer. This is achieved by sending m sequential one-hop messages hm_i as described for the accusation messages am, with $hm_i = (f_{K_{ij}}(Q_i, ID_i, revoke), (Q_i, ID_i, revoke))$. The recipients verify hm_i, and if successful update their $\mathcal{KRL}$ with $c_i = 1$ and $a_{i,i} = 1$.

Finally, we need to consider newly joining nodes ID_n. Every ID_n needs to create its own $\mathcal{KRL}_n$, which ideally contains all past accusations for all unexpired keys in m-$\mathcal{N}_n$. Therefore, ID_n sends a ‘hello’ message to its one-hope neighbors with $h = (ID_n, r_n, hello)$, where r_n is a random nonce to prevent replay attacks. The recipients compute Q_n from ID_n and the currently used expiry date. Latter is chosen in publicly known intervals as previously discussed. Note that Q_n needs only to be transmitted if it is in form of Equation 3 and $v > 1$. In that case it needs to be verified if Q_n corresponds to ID_n. Next, each node ID_i in $\mathcal{N}_n$ computes K_{in} and sends a welcome message $wm_i = (f_{K_{in}}(ID_i, e_i, \mathcal{KRL}_i, r_n + 1), ID_i, e_i, \mathcal{KRL}_i, r_n + 1)$ back to ID_n, where $e_i = |\mathcal{N}_i|$. ID_n does not consider information from $\mathcal{KRL}_i$ to construct its own revocation list under the following conditions: (1) wm_i cannot be verified successfully; (2) Q_i is marked as revoked by any of the received $\mathcal{KRL}$; (3) $e_n \ll e_{av}$, where e_{av} is the average of all received neighborhood sizes e_i and the e_n is the number of received welcome messages; and (4) $\mathcal{KRL}_i$ significantly differs from the majority of other received lists. Basically, ID_n is looking for a majority agreement of accusations. If the received lists do not give such a majority or the number of lists is not sufficient ($e_n \ll e_{av}$), ID_n sends the hello message to its 2-hop neighbors. This might be repeated m times, however, after several tries, ID_n should consider joining the network somewhere else, because it obviously joined a bad neighborhood. We would like to stress, that in a normal environment the received data should be fairly consistent.

4.5 Fully Self-Organized MANET-IDAKE scheme

In the basic MANET-IDAKE scheme, we assume two network phases, namely the initialization phase with access to an external KGC and the running system phase without access to a KGC. However, we now consider applications where no central external KGC is available at any time. For this reason we propose a *fully self-organized MANET-IDAKE* scheme. The scheme is well-suited for all applications that require all algorithms to be executed within the network, i.e. independent of a TTP or any other infrastructure, even at the time the network is formed. To do so, we need to emulate an internal KGC. We adopt the ID-based (k, n)-threshold schemes that have been proposed for MANETs [13, 20] for that matter. Both schemes provide suitable Algorithms 1-3 which can be used in our fully self-organized MANET-IDAKE, whereas Algorithm 4. can be adopted from the basic MANET-IDAKE scheme. However, there is no existing renewing or revocation algorithm for fully self-organized ID-based schemes, neither for MANET nor for general networks. For key renewing, we use the extract and distribute algorithms from [13, 20] with a public key format as given in Equation 3. In that way keys can be renewed in the network by an internal KGC. For the revocation, i.e. Algorithm 6, the algorithm introduced for the basic MANET-IDAKE can be adopted. This is possible because the algorithm can be executed independently of any KGC, internal or external, and runs fully self-organized in the network.

4.6 Lightweight IDAKE Protocol

We present a lightweight IDAKE protocol to provide secure and efficient authentication and key exchange for two network nodes ID_i and ID_j. The protocol is identical to the *REKEY* protocol in [10] with the only difference, that it uses pre-shared keys K_{ij} from Equation 2 as the required pre-shared secrets. The pre-shared keys are used as input of a secure and publicly known MAC function $f(\cdot)$ to derive two different keys, namely an authentication key $k_a = f_{K_{ij}}(1)$ and a key derivation key $k_d = f_{K_{ij}}(2)$. Refer to Protocol 1 for the protocol flow, where s is a session identifier and N_i and N_j are random nonces, and see [10] for more details on the computation steps.

Protocol 1. Lightweight IDAKE protocol

Pre-shared Key: $k_a = f_{K_{ij}}(1), k_d = f_{K_{ij}}(2)$

Protocol Flow:

1. $ID_i \longrightarrow ID_j : ID_i, s, N_i$
2. $ID_i \longleftarrow ID_j : ID_j, s, N_j, r_j = f_{k_a}(ID_i, N_i, s, N_j)$
3. $ID_i \longrightarrow ID_j : ID_i, s, r_i = f_{k_a}(ID_j, N_j, s, N_i)$

Session Key: $SK = f_{k_d}(N_i, N_j)$

4.7 Extensions

To offer more security features in the revocation, authentication, key exchange, and other security protocols in the presented MANET-IDAKE schemes, new algorithms need to be introduced. We introduce several possible extensions below. Please note that some of these extended algorithms require the setup algorithm to be modified in a way, that for instance more hash functions are chosen and part of the public parameters. Please refer to [4, 7, 15] for details.

To enable non-repudiable communications and one-to-many authentications, we need to implement Algorithms *(7.) Sign* and *(8.) Verify*, e.g. [15]. First feature can be desirable in IDAKE protocols, whereas later can enable the broadcast of signed harakiri or accusation messages. To achieve perfect forward secrecy (PFS) in an IDAKE protocol an Algorithm *(9.) ECDH* to compute the session key according to elliptic curve DH is required. Other schemes might need Algorithms *(10.) Encrypt* and *(11.) Decrypt*, e.g. from [7], in addition to the basic scheme. However, we would like to stress that encryption under a session key derived by executing Protocol 1 is much more efficient. To enable immediate threshold-based key revocation in fully self-organized MANET-IDAKE schemes, an Algorithms *(12.) Threshold Sign* needs to be implemented, for instance [4]. This would enable a group of k nodes to collaboratively revoke a key and broadcast the signed revocation message. However, threshold signing schemes are computationally challenging and create a large overhead.

5 Security and Performance Discussion

Efficient and secure implementations of IBC schemes and bilinear parings are introduced in the literature, e.g. [3, 5, 7, 25]. It has been shown that those schemes are feasible for implementation on very constrained platforms such as smartcards [5]. In implementations that provide a 1024-bit RSA security level, a 512 bit curve is chosen, and the computations are executed in 170 bit subgroups. Consequently, storage requirements are very low and clearly outperform RSA and compete with ECC implementations. Nodes can either store pre-shared keys or public keys together with the corresponding identities or derive the keys from the identities every time they are needed. This constitutes a memory/computation trade-off. However, computing pre-shared keys requires a pairing computation and computing public keys applying the mapping function $H_1(\cdot)$. On the other hand, memory space is very cheap and growing fast, while the pre-shared keys and public keys are fairly short. Furthermore, the presented schemes only require the storage of keys from an m-hop neighborhood as opposed to storage of all keys in the network. Hence, memory resources can be assumed to be sufficient for key storage in most applications.

Besides the low memory requirements, the MANET-IDAKE schemes have low bandwidth requirements due to the efficient key management of IBC schemes. In addition, the presented revocation schemes make use of one-hop or m-hop message forwarding instead of full network broadcasts as suggested for revocations in PKI schemes [12, 22]. The reduced propagation range significantly reduces the overall network communication overhead and lowers the energy consumption of single nodes.

The computational complexity of the presented MANET-IDAKE schemes depends on the implemented key revocation and renewal algorithms and the used IDAKE protocol. The algorithms and protocol all require the computations of pre-shared keys K_{ij} in form of Equation 3, which in turn requires the computation of one bilinear pairing. Clearly, the pairing computations are the only demanding computations in the presented schemes. Weil and Tate pairing are suitable pairings, where latter is favored due to its better computational performance. Efficient implementations exist, e.g. [3, 5, 25], and the Tate pairing has been successfully implemented on a smartcard [5]. These results show that pairing computations are feasible in even very constrained environments such as MANETs. We would like to emphasize that the pairing computation is only needed the first time two nodes communicate with each other. The computed pre-shared key can then be used in several algorithms and protocols in different network layers. The use is not only limited to the presented revocation algorithms and IDAKE protocols, e.g. the pre-shared keys K_{ij} could be also used in the Destination-Sequenced Distance-Vector (SEAD) routing protocol [17]. Hence, despite the fairly high cost of the first time key computation, the overall costs are significantly reduced due to the frequent re-use of these keys. Once the keys are computed, all other computations are symmetric primitives such as MAC computations.

We now briefly discuss the security and performance of the neighborhood watch scheme, compare the two presented MANET-IDAKE schemes, and analyze Protocol 1.

Neighborhood watch scheme: In this scheme, trust is based on monitoring neighbors and one-hop trust relationships. Node ID_i which trusts a neighbor ID_j, also trusts that this neighbor is capable of observing its own neighbors, and thus maintaining a correct accusation list. If one of the intermediate hops of a multi-hop routing path fails to detect a misbehaving node, the trusted path is broken. The scheme is extremely efficient because it only requires one-hop messages and the computation of MAC functions, whereas other proposed schemes require broadcasts and signing. It could be argued that each one-hop message in the presented schemes requires the computation of one pairing if two nodes never communicated before, which can cause a poor overall performance. However, this is very unlikely in a fairly static neighborhood or by implementing a routing protocol that prefers known neighbors. We would like to stress that the presented approach differs from PGP, because a pair of nodes ID_i and ID_j already shares an authentic key K_{ij} that is derived from key material previously issued by a trusted KGC. Our scheme serves to check wether an authentic key is compromised, a check that is not provided in PGP schemes. However, the presented neighborhood watch scheme is not a revocation scheme since accusations are not forwarded at all and harakiri messages only to the one-hop neighborhood. As a consequence, attacks by roaming adversaries cannot be prevented. Nevertheless, the scheme is suitable for static ad hoc networks or as sub scheme of revocation schemes for MANETs.

Basic MANET-IDAKE scheme: In the basic scheme, we assume that a KGC is available to set up the nodes before they join the network, i.e. the KGC executes the set-up, extract, and distribute algorithms. Consequently, the network initialization phase does not require any computations by the nodes or communication in the network and is hence very efficient. We believe that this scenario is very likely in most applications. For instance, devices might be initialized by the manufacturer, network provider, or system administrator. The key revocation needs to be handled internally and is thus the only algorithm that creates a network overhead by sending one-hop or m-hop messages. For this reason, we suggest evaluating whether revocation can be omitted. This is a good approach in all MANETs with short lifetime or applications where frequent key renewals are possible.

In the revocation algorithm the security parameter for newly joining nodes is not the threshold δ, the security rather depends on the majority computation of the $\mathcal{KRL}$. However, to achieve δ security throughout the network, each accusation would need to be signed and stored together with its signature in the $\mathcal{KRL}$. In that case, MACs cannot be used, because the accusations need to be verifiable by every network node. Hence, a newly joining node would need to verify every single accusation in all received revocation lists. This requires a maximum of eN^2 verifications, which is clearly too expensive not only for MANET applications. We conclude that accusation schemes are only feasible if the entire list is securely distributed instead of securing each individual accusation.

Fully self-organized MANET-IDAKE scheme: Clearly, the fully self-organized scheme shows worse performance in terms of computational and communication costs than the basic scheme. This is due to the use of threshold schemes to emulate an internal KGC which creates a large communication and computational overhead during the network initialization. However, an advantage of the scheme is that key renewing can be performed within the network. In that case it needs to be evaluated whether it is more efficient to replace key revocation by frequent key renewals.

IDAKE protocol: As mentioned earlier, Protocol 1 has the same message flows as the *REKEY* protocol in [10]. The novelty of Protocol 1 is to provide a way to securely establish the required pre-shared secret in a non-interactive way. The REKEY protocol has been proven to be SK-secure without PFS in the Canetti-Krawczyk security model [10]. The pre-shared keys K_{ij} have been shown to be secure for usage as MAC-based authenticators [8]. Both proofs combined let us conclude that Protocol 1 is SK-secure without PFS if the following three conditions hold: (1) the pre-shared keys K_{ij} from Equation 2 are random keys chosen under security parameter k; (2) the bilinear DH problem is hard (as defined in [7]) in the implemented MANET-IDAKE scheme; and (3) the employed MAC $f(\cdot)$ is secure. Although Protocols 1 does not provide PFS, we believe that Protocol 1 is attractive for all applications where long-term secrecy of data is not required and authenticity is more important than the secrecy of the data. However, if PFS is inevitable in the IDAKE schemes, we refer to an IDAKE protocol in [8], which is essentially Protocol 1 in which the symmetric key derivation function is replaced by a DH-like key agreement.

Protocol 1 shows excellent computational and communication performance and is thus attractive for an implementation in MANETs. In the first protocol execution between two nodes ID_i and ID_j, Protocol 1 requires 1 pairing computation and 2 MAC computations. For all consecutive protocol executions between the same nodes, Protocol 1 uses purely symmetric primitives. Many of the common attacks on (ID)AKE protocols, such as impersonation, session key, identity misbinding and other attacks on the session key are prevented by Protocol 1. However, to prevent key compromise impersonation attacks [8] or provide PFS, the protocol would need to employ ID-based signature or encryption schemes, and/or ECDH key agreements. However, using one or more of these schemes would significantly increase the computational complexity of the protocol.

6 Conclusions and Future Work

In this paper we show that pairing-based IBC schemes are an attractive alternative to existing schemes for securing ad hoc networks. IBC schemes offer efficient key management and the pre-shared secret keys enable the use of symmetric cryptography. We utilize these features to design two IDAKE schemes that are suitable for MANETs. We are the first to describe a fully functional IBC scheme for MANETs, including the format of the used identities, network initialization, key renewal, key revocation, authentication, and key exchange. Especially key renewal and revocation algorithms have never been proposed for IBC schemes before. We introduce a fully self-organized revocation algorithm to revoke ID-based public keys in a MANET and show how public keys can be renewed for the same identities. Furthermore, we present a secure lightweight IDAKE protocol. The first MANET-IDAKE scheme is very efficient and suitable for applications in which an external KGC is available during the network initialization phase, whereas the second scheme is fully self-organized for sacrificing performance.

We plan to further enhance the presented schemes in the following ways: (1) integrating secure routing to increase the overall performance by using pre-shared keys, e.g. in [17]; (2) designing more key revocation and renewing algorithms, e.g. for networks with sporadic backbone access, such as mesh networks, or networks that allow threshold signing; and (3) designing and analyzing more IDAKE protocols that offer more security features than Protocol 1. Whereas most of these extensions have been already designed and are part of a separate work, we also plan to analyze the security/performance trade-off of the presented m-hop MAC-based revocation scheme compared with a sign&broadcast approach in different network settings.

References

[1] N. Asokan and P. Ginzboorg. Key Agreement in Ad Hoc Networks, *Computer Communications*, vol. 23, no. 17, 2000, pp. 1627-1637.

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

[3] P.S.L.M. Barreto, B. Lynn, and M. Scott. Efficient Implementation of Pairing-Based Cryptosystems, *Journal of Cryptology*, vol. 17, no. 4, 2004, pp. 321-334.

[4] J. Baek and Y. Zheng. Identity-Based Threshold Signature Scheme from the Bilinear Pairings, *International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 1, pp. 124-128, 2004.

[5] G.M. Bertoni, L. Chen, P. Fragneto, K.A. Harrison, and G. Pelosi. Computing Tate Pairing on Smartcards. Available at `http://www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf`.

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

[7] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing, *Advances in Cryptology - CRYPTO '2001*, LNCS 2139, pp. 213-229, 2001.

[8] C. Boyd, W. Mao, and K.G. Paterson. Key Agreemet Using Statically Keyed Authenticators, *Applied Cryptography and Network Security, ACNS 2004*, LNCS 3089, pp. 248-262, 2004.

[9] M. Cagalj, S. Capkun and J.P. Hubaux. Key Agreement in Peer-to-Peer Wireless Networks, to appear in *Proceedings of IEEE, Special Issue on Security and Cryptography, 2005.*

[10] R. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels, *Advances in Cryptology - EUROCRYPT '01*, LNCS 2045, pp. 453-474, 2001. Full version available at the Cryptology ePrint Archive: Report 2001/040.

[11] L. Chen and C. Kudla. Identity Based Authenticated Key Agreement Protocols from Pairings. *Hewlett-Packard Technical Report HPL-2003-25 20030212*, HP Laboratories, 2003.

[12] C. Crépeau and C.R. Davis. A Certificate Revocation Scheme for Wireless Ad Hoc Networks. *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, ACM Press, isbn 1-58113-783-4, pp.54-61, 2003.

[13] H. Deng, A. Mukherjee, D.P. Agrawal. Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks, *International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 1, pp. 107-115, 2004.

[14] 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.

[15] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. *Selected Areas in Cryptography –SAC 2002*, LNCS 2595, pp. 310-324, 2003.

[16] K. Hoeper and G. Gong. Short paper: Limitations of Key Escrow in Identity-Based Schemes in Ad Hoc Networks, *Security and Privacy for Emerging Areas in Communication Networks (SecureComm 05)*, 2005.

[17] Y.C. Hu, D.B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad-Hoc Networks, *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, 2002, pp. 3-13.

[18] 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.

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

[20] A. Khalili, J. Katz, and W.A. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks, *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, IEEE Computer Society, pp. 342-346, 2003.

[21] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Secure Key Issuing in ID-Based Cryptography, *CRPIT '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, Australian Computer Society, Inc., 2004, pp. 69-74.

[22] 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.

[23] J. Luo, J.-P. Hubaux, and P.Th. Eugster. DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks, *EPFL Technical Report IC/2004/106*, School of Computer and Communication Sciences EPFL (Swiss Federal Institute of Technology), to appear in IEEE Transactions on Dependable and Secure Computing (TDSC).

[24] B. Lynn. Authenticated identity-based encryption, *Cryptology ePrint Archive*, Report 2002/072, 2002.

[25] V.S. Miller. The Weil Pairing, and Its Efficient Calculation, *Journal of Cryptology*, vol. 17, no. 4, 2004, pp. 235-261.

[26] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairings, *The 2000 Symposium on Cryptography and Information Security*, 2000.

[27] A. Shamir. Identity-based Cryptosystems and Signature Schemes, *Advances in Cryptology - CRYPTO 84*, pp. 47-53, 1985.

[28] 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*, LNCS 1796, Springer, pp. 172-194, 1999.

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