Key Revocation for Identity-Based Schemes in Mobile Ad Hoc Networks

Katrin Hoeper[1] and Guang Gong[2]

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, N2L 3G1, Canada
[1]khoeper@engmail.uwaterloo.ca and [2]ggong@calliope.uwaterloo.ca

Abstract. Recently, identity-based cryptographic (IBC) schemes have been considered to secure mobile ad hoc networks (MANETs) due to their efficient key management properties. However, proposed schemes do not provide mechanisms for key revocation and key renewal. In this paper, we propose the first key revocation and key renewal mechanisms for IBC schemes that are especially designed for MANETs. In our fully self-organized revocation scheme, each node monitors nodes in communication range and securely propagates its observations. The public key of a node is revoked if a minimum number of nodes accused the node. To enable key renewal, we introduce a modified format for ID-based public keys, such that new keys can be issued for the same identity. The introduced revocation scheme is efficient because it uses pre-shared keys from the Weil pairing to secure accusation and revocation messages and messages are sent to an m-hop neighborhood instead of to the entire network. Our revocation mechanism can be adapted to PKI schemes in MANETs.

1 Introduction

A growing number of mobile wireless applications require that networks are spontaneously formed by the participating 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 implementing security is of paramount importance in MANETs. However, the special properties of MANETs, such as the lack of infrastructure including the 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. Due to the problem of secure key distribution in symmetric schemes, the use of a public key infrastructure (PKI) is desirable in many MANET applications. The major challenges of implementing PKIs in MANETs are issuing and distributing certificates and enabling certificate revocation. Many PKI-based schemes have been proposed to secure MANETs, e.g. [3, 9, 11], whereas many of them do not provide mechanisms for certificate revocation at all, e.g. [11]. In general, one can distinguish two PKI implementations in MANETs. In the first one, a certification authority (CA) issues public key certificates to nodes before the nodes join the network, we call this an *off-line CA*. In the second case, nodes obtain their certificates from a group of network nodes that serve as *distributed on-line CA*. Distributed CAs can be implemented using (k, n)-threshold schemes [9,11]. Implementations using on-line TTPs have the advantage of not requiring the set-up of any infrastructure. However, the threshold scheme imposes a lot of communication and computational load onto the network.

Recently, identity-based cryptographic (IBC) schemes have been considered as an alternative public key scheme to secure MANETs due to their efficient key management [4,6,8]. However, the proposed IBC schemes do not provide mechanisms for key revocation and key renewal. We believe that these mechanisms are of paramount importance and every node in a MANET should be able to instantly verify whether a public key has been revoked. Due to the weak physical protection of nodes, node compromises including key disclosures are very likely in MANETs. Frequent key renewals to prevent such compromises are either computationally challenging in solution with distributed on-line key generation center (KGC) or infeasible in solutions with off-line KGC. In this paper, we propose the first key revocation and key renewal schemes for IBC schemes that are especially designed to meet the requirements and constraints of MANETs. In our revocation scheme, each node uses a neighborhood watch scheme to monitor nodes in communication range for suspicious behavior. These observations are then securely propagated to an m-hop neighborhood. The public key of a node is revoked if at least δ nodes accused the node. Our key revocation scheme is scalable in parameters m and δ, i.e. the level of security can be chosen as performance trade-off. To enable key renewal in IBC schemes, we introduce a new format for ID-based public keys such new keys can be issued for the same identity after the previous key has been revoked.

The proposed key revocation scheme can be adapted to PKI-based solutions for MANETs to provide certificate revocation. Other than existing certificate revocation schemes for MANETs, e.g. [3,9], our scheme reduces the overall computational and communication network overhead by using pre-shared keys from the Weil pairing to secure accusations, and sending messages to an m-hop neighborhood rather than to the entire network. Other than in existing schemes, we discuss and efficiently solve the problems of nodes that wish to revoke their own keys and new nodes that join the network and wish to learn about past accusations and revocations.

The remainder of the paper is organized as follows. In the next section we discuss the system set-up for our key revocation and key renewal schemes for IBC-based MANETs. The revocation and renewal schemes are then introduced in Sect. 3. The security of the scheme is analyzed in Sect. 4 and compared to related work in Sect. 5. Finally, we discuss the contributions of the proposed schemes in Sect. 6.

2 System Set-up

2.1 Preliminaries

In this paper, we present key revocation and key renewal schemes for IBC implementations that are based on IBC schemes from the Weil pairing [1]. In the remainder of the paper, we adopt many notations from [1], please refer to the original paper for details. IBC schemes provide a very efficient key management that helps reducing communication, computation, and memory costs. A summary of desirable properties of IBC scheme for implementations in MANETs can be found in [6]. The main feature of IBC schemes is the use of self-authenticating public keys, which makes the use of public key certificates redundant. Because the public key Q_i of a node ID_i is pre-determined, the private key d_i is derived from Q_i and a master secret key s that is only known to the KGC, i.e. $d_i = sQ_i$. The KGC generates and distribute the private keys during the initialization of all network nodes. In IBC schemes, every network node is able to derive the public key Q_i of a communication partner ID_i in the network, e.g. $Q_i = H_1(ID_i)$ [1]. This does not require the exchange of any data. In addition, 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 $K_{i,j}$ [2] in a non-interactive fashion as given in (1).

$$K_{i,j} = \hat{e}(d_i, Q_j) = \hat{e}(Q_i, d_j) \tag{1}$$

For the key computation, both nodes compute the bilinear mapping $\hat{e}(\cdot)$ over their own private key d_i and the public key Q_j of the other node.

The KGC is a key escrow because it knows all private and pre-shared keys in the network. Threshold schemes for distributed KGCs have been introduced for this matter [1]. Other preventions are known as well, e.g. the limited power of key escrows in MANETs has been analyzed in [5].

2.2 Choosing Identities

Identities ID_i must be *unique* for each entity i 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. Generally, we can distinguish three cases of entities an identity can be 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, user-dependent IDs seem desirable. In sensor networks or other MANETs in which user do not operate the devices, the MAC address seems to be a good choice. 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 are dynamic or do not exist at all.

2.3 Public Key Format

To limit the validity period of an ID-based public key, an expiry date can be embedded in the key itself, e.g. $Q_i = H_1(ID_i||t_i)$, where the expiry date t is concatenated with the identity [1]. Only a node that is in possession of the private key d_i which corresponds to the date t_i can sign or decrypt messages. However, this key format is only sufficient in schemes without revocation. In schemes with explicit key revocation, i.e. not only implicit revocation by key expiry, nodes need to be able to request key renewal even before the expiry date t. The key renewal is required in the case of key compromise or revocation. Since the identity ID_i is unchangeable in IBC schemes as discussed in Sect. 2.2, issuing a key for the same expiry date would result into the same old compromised key. However, issuing new keys with a new expiry date t' might not be feasible, because a node ID_i is only eligible to possess keys until t. Furthermore, it is desirable in IBC schemes that expiry dates are chosen in a predictable manner, e.g. in fixed intervals, such that nodes do not need to exchange public keys after the previous keys expired. Hence, to provide immediate key renewal after key compromise, we need to add some additional data v to the public key that can be changed with every key renewal. We use the following format as given in (2) below, where v is the version number of the public key.

$$Q_i = H_1(ID_i||t_i||v_i) \tag{2}$$

For instance, upon compromise of $Q_i = H_1(ID_i||t_i||v_i)$, node ID_i can request a new key Q'_i before t_i, with $Q'_i = H_1(ID_i||t_i||v'_i)$ and $v'_i = v_i + 1$. Note that the version number v always starts with $v = 1$ for every new expiry date t and is incremented with each key renewal for the same date t. Key renewal and distribution with $v > 1$ are discussed in Sect. 3.3.

2.4 Assumptions

The necessary assumptions for the network and its nodes in our IBC key revocation and key renewal schemes can be summarized as follows: *1. bidirectional communication links*; *2. nodes are in promiscuous mode*; *3. each node has a unique identity*; *4. nodes know identities of their one-hop neighbors*; *5. nodes know distances to other nodes in m-hop neighborhood*; and *6. nodes obtain a private and public key pair from an off-line KGC prior joining the network.*

The first two assumptions are necessary to enable nodes to monitor their neighbor nodes in communication range. This is required in our revocation scheme. Bidirectional links are a common assumption in many lower-layer MANET protocols, e.g. in the AODV [10] and other AODV-based routing protocols. Promiscuous mode is assumed in dynamic routing protocols for MANETs, e.g. AODV and DSR [7]. Assumption 3 is necessary to unambiguously identify nodes. This kind of identifiers are required for many network tasks and protocols, such as routing and authentication. Assumption 4 is needed, because neighbor nodes need to be unambiguously identified before they can be marked as suspicious or trustworthy in the revocation scheme. This information is usually provided by routing and other lower-layer protocols, e.g. AODV. In case the identities of neighbors are not provided by lower layer protocols, users first explore their neighborhood by sending *hello* messages and waiting for the responses. Assumption 5 is necessary to enable nodes to decide which accusation values they need to consider for updating their revocation lists, e.g. accusations from nodes more than m hops away are discarded. This information is provided by the routing protocols, e.g. AODV and DSR. Assumption 6 is necessary because cryptographic keys are needed to provide message authentication in our revocation scheme. Here we assume an external off-line KGC that issues the keys. The assumption can be changed to a distributed on-line KGC that issues keys within the network as outlined in Sect. 5. We can summarize that the necessary assumptions are quite common, and in fact necessary in most ad hoc routing and security protocols. Hence, our assumptions do not impose an additional burden to the system.

3 Key Revocation and Renewal for IBC Schemes in MANETs

3.1 Key Revocation

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. As discussed earlier, 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 our scheme, keys are revoked either if a node notices that its own key has been compromised or if a group of at least δ nodes observes that another node behaves suspiciously.

In order to provide key revocation in IBC schemes for MANETs, we introduce three algorithms. First, nodes observe the nodes in their neighborhood for suspicious behavior using *Algorithm 1: Neighborhood watch.* Second, nodes need to be able to revoke their own public keys using *Algorithm 2: Harakiri.* Third, nodes securely inform each other about suspicious observations and generate key revocation lists in *Algorithm 3: Accusation scheme.*

Alg.1 Neighborhood watch: The neighborhood watch scheme is a local monitoring scheme, in which each node ID_i monitors its one-hop neighborhood $\mathcal{N}_i$ for suspicious behavior. Suspicious behavior can be frequent packet drops or a large number of sent messages. Therefore nodes observe their neighbors and check for instance whether the nodes forwarded messages that were addressed to another node. Suspicious behavior can have different causes, e.g. a node has been compromised and is now controlled by a malicious user, or a node is selfish and rather conserves its energy than forwarding messages.

For an easier representation and without loss of generality, we denote ID_i's one-hop neighbors as $ID_j \in \mathcal{N}_i$ with $j \in \{1, \dots, N_i\}$, where N_i is the number of one-hop neighbors. User ID_i stores so-called accusation values $a^i_{i,j}$ for each $ID_j \in \mathcal{N}_i$ together with the expiry date t^i_j and version number v^i_j of the current public key Q_j. A node ID_i sets its accusation values $a^i_{i,j} = 1$ if ID_i observed ID_j to behave suspiciously, otherwise $a^i_{i,j} = 0$. The accusation values that a node ID_i creates from its neighborhood watch, can be represented as an accusation matrix

$$AM^i = \begin{pmatrix} ID_1 & (t^i_1, v^i_1) & a^i_{i,1} \\ \vdots & \vdots & \vdots \\ ID_{N_i} & (t^i_{N_i}, v^i_{N_i}) & a^i_{i,N_i} \end{pmatrix} \text{with } a^i_{i,j} \in \{0,1\} \text{ and } j \in \{1, \dots, N_i\}.$$

Each row vector $\underline{r}^i(ID_j)$ in AM^i, we use $\underline{r}^i_j$ for short in the rest of the paper, corresponds to a neighbor $ID_j \in \mathcal{N}_i$ and the accusation values $a^i_{i,j}$ for the current public key Q_j with expiry date t^i_j and version number v^i_j. We refer to the third column in AM^i as column vector $\underline{c}^i_i$, which is the vector that contains all accusations made by ID_i. The accusation values are updated every time ID_i observes suspicious behavior. Once the flag $a^i_{i,j}$ is set, the value will not be reset to zero until a new public key Q'_j is received.

Alg.2 Harakiri: When a node ID_i realizes that its private key d_i has been compromised, it broadcasts a harakiri message hm_i, with

$$hm_i = (ID_i, d_i, Q_i, (t_i, v_i), \text{“revoke”}, \textit{hopcount}),$$

to its m-hop neighborhood m-$\mathcal{N}_i$. Node ID_i initially sets $\textit{hopcount} = m$ and sends the message to all one-hop neighbors $\mathcal{N}_i$. The receivers ID_j verify if the harakiri is authentic, by checking wether (3) is true.

$$K_{j,i} = \hat{e}(d_i, Q_j) \tag{3}$$

The check verifies whether the broadcasted private key d_i really corresponds to the public key Q_i. Therefore, a recipient of hm_i, say node ID_j, looks up whether it is in possession of the public key Q_i and the pre-shared key $K_{j,i}$ and if so, uses the $K_{j,i}$ to check whether (3) is true. If ID_j is not in the possession of these keys, ID_j first computes Q_i according to (2), to check whether ID_i and Q_i correspond to each other. If this check is successful, ID_j derives $K_{j,i}$ according to (1) and then checks whether (3) is true. If (3) is true, the receiver ID_j updates its accusation value $a^j_{i,i} = 1$, decrements *hopcount* and broadcasts the message again. Otherwise, hm_i is discarded. The broadcasting is repeated until $\textit{hopcount} = 0$. This ensures that all nodes in an m-hop neighborhood of the compromised node ID_i receive the harakiri message and thus learn about the key compromise.

Alg.3 Accusation scheme: In this algorithm every node ID_i creates its own key revocation list $\mathcal{KRL}^i$ for its m-hop neighborhood. In this paragraph we will describe how revocation lists are created (Alg.3.1), securely propagated (Alg.3.2) and how nodes use received revocation lists and harakiri messages to update their own revocation lists (Alg.3.3).

Alg.3.1 Creating a key revocation list $\mathcal{KRL}$: Each node ID_i creates a key revocation list $\mathcal{KRL}^i$ of the following format:

$$\mathcal{KRL}^i = \begin{pmatrix} ID_1 & (t_1^i, v_1^i) & R_1^i & a_{1,1}^i & \cdots & a_{1,M_i}^i \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ ID_{M_i} & (t_{M_i}^i, v_{M_i}^i) & R_{M_i}^i & a_{M_i,1}^i & \cdots & a_{M_i,M_i}^i \end{pmatrix},$$

with $a_{j,k}^i \in \{0, 1, -\}$ and $j, k \in \{1, \ldots, M_i\}$, where M_i is the number of nodes in ID_i's m-hop neighborhood m-$\mathcal{N}_i$, including ID_i itself. The accusation value $a_{j,k}^i = -$ indicates that ID_j and ID_k are more than m hops away from each other, and thus cannot give a statement about each others trustworthiness. Each row vector $\underline{r}^i(ID_j)$, short $\underline{r}_j^i$, in $\mathcal{KRL}^i$ corresponds to a node $ID_j \in m$-$\mathcal{N}_i$, where the row contains the accusation values $a_{j,k}^i$ from all nodes $ID_k \in m$-$\mathcal{N}_i$ against a node ID_j. Each column vector $\underline{c}^i(ID_j)$, short $\underline{c}_j^i$, in $\mathcal{KRL}^i$ contains all the accusations $a_{k,j}^i$ made by node ID_j against all nodes $ID_k \in m$-$\mathcal{N}_i$. The index ^{i} denotes that the values are the current values in ID_i's $\mathcal{KRL}^i$, where other nodes might have different values. For example, $a_{j,k}^i \neq a_{j,k}^l$ for $i \neq l$ in some cases. Discrepancies in accusation values can exist, because accusation values may be more or less up to date, and nodes have different m-hop neighborhoods, and thus receive different accusation and harakiri messages.

The first field in each row $\underline{r}_j^i$ in $\mathcal{KRL}^i$ contains the identity of node ID_j, the second field the expiry date t_j^i and version number v_j^i of the most recent public key Q_j that ID_i knows of. The fields $4-(M_i+3)$ contain the accusation values $a_{j,1}^i$ - a_{j,M_i}^i, where value $a_{j,k}^i = 1$ indicates that node ID_k accused node ID_j, and $a_{j,k} = 0$ otherwise. The third field contains a 1-bit flag R_j^i that, when set, indicates that node ID_i considers the public key Q_j of node ID_j as revoked. The revocation flags R_j^i in ID_i's key revocation list $\mathcal{KRL}_i$ are set, i.e. $R_j^i = 1$, if one of the following four conditions is true:

(Cond.1): $a_{j,i}^i = 1$, i.e. node ID_i observed itself the malicious behavior of node ID_j during the neighborhood watch (Alg.1). This follows that ID_j and ID_i are 1-hop neighbors.

(Cond.2): t_j^i is expired, i.e. the current copy of the ID_j's public key Q_j is expired.

(Cond.3): $a_{j,j}^i = 1$, i.e. ID_i received an authentic harakiri message hm_j from ID_j.

(Cond.4): $A_j^i = \sum_{k=1}^{M_i} a_{j,k}^i > \delta$ for all k, s.t. $R_k^i = 0$, i.e. add all accusation values $a_{j,k}^i$ of row vector $\underline{r}_j$ from non-revoked nodes ID_k and check whether the sum is greater than δ. In other words, the public key Q_j is revoked if node ID_i received more than δ accusations from trustworthy nodes ID_k for a suspicious node ID_j. Note that "-" is treated as zero value in the sum.

If none of the four conditions applies, $R_j^i = 0$, i.e. ID_j and its current public key Q_i are considered to be trustworthy.

Alg.3.2 Propagating accusations: In this algorithm nodes securely propagate accusations through the network. Every time a node ID_i updates its accusation matrix AM^i because it observed some suspicious behavior in its neighborhood watch, ID_i sends an accusation message to its one-hop neighbors. Similarly, every time ID_i updates its key revocation list $\mathcal{KRL}^i$ because it received accusations from other nodes, ID_i sends an accusation message to its neighbors. The accusation messages send by ID_i have the following format :

$$am_{i,j} = (f_{K_{i,j}}(ID_i, am_i), (ID_i, am_i)), \text{ for all } ID_j \in \mathcal{N}_i$$

where $am_i = AM^i$ for updates from the neighborhood watch of ID_i, and $am_i = \mathcal{KRL}^i$ for updates of the revocation list caused by accusation messages that ID_i received from other nodes. Optionally, am_i contains only the updated values to reduce bandwidth. The accusation messages $am_{i,j}$ are secured using the pre-shared keys $K_{i,j}$ for all $ID_j \in \mathcal{N}_i$ and then unicasted to each one-hop neighbor ID_j. The pre-shared keys serves as input in a secure hash function $f(\cdot)$ to authenticate the message. Upon receiving $am_{i,j}$, a neighboring node ID_j verifies the received message using its pre-shared key $K_{j,i}$. If the verification is successful, ID_j updates its key revocation list $\mathcal{KRL}^j$ accordingly, as we will explain in the next paragraph.

Alg.3.3 Updating key revocation lists: Every time a node ID_i receives a harakiri message hm_j from ID_j, the node verifies the message as described in Alg. 2 and if this verification is successful, node ID_i sets $a^i_{j,j} = 1$ and thus $R^i_j = 1$ in its revocation list $\mathcal{KRL}^i$ (see Cond. 3 in Alg. 3.2). If a node ID_i receives an accusation message $am_{j,i}$ of an one-hop neighbor ID_j, node ID_i performs the following steps to update its key revocation list $\mathcal{KRL}^i$:

(Step 1): check whether $R^i_j = 0$, i.e. whether ID_i considers ID_j to be trustworthy; if yes continue, else discard $am_{j,i}$ and stop.

(Step 2): verify authenticity of $am_{j,i}$ using the pre-shared key $K_{i,j}$ as described in Alg. 3.2; if verification is successful continue, else discard $am_{j,i}$ and stop.

(Step 3): extract column vector $\underline{c}^j_j$ of AM^j or $\mathcal{KRL}^j$ from $am_{j,i}$ to update column vector $\underline{c}^i_j$ in $\mathcal{KRL}^i$, i.e. adopt the accusation values from ID_j's neighborhood watch. Note that ID_i uses only accusation values that are addressed to nodes in ID_i's own m-hop neighborhood, other accusation values are discarded. If $am_{j,i} = AM^j$ stop, else continue.

(Step 4): discard all columns $\underline{c}^j_k$ from $\mathcal{KRL}^j$ for: $k = i$ because that is ID_i's own accusation vector; $k = j$ because that one was used in step 3; $k = l$ for all $R^i_l = 1$ with $l \in \{1, \ldots, M_i\}$ because ID_i does not trust nodes ID_l; $k = r$ for all $am_{r,i}$ that were accepted in step 2; $k = s$ for all $ID_s \notin m\text{-}\mathcal{N}_i$, i.e. nodes that are more than m hops away. Save all other columns $\underline{c}^j_k$.

Now ID_i repeats steps 1-4 for all received accusation messages $am_{j,i}$. Lets say ID_i saved d_k column vectors $\underline{c}^j_k$ from d_k different nodes ID_j for the same ID_k in step 4. Now for every $d_k > \varepsilon$, with ε being a security threshold, ID_i performs step 5 below.

(Step 5): use all d_k saved column vectors $\underline{c}_k^j$ from step 4 to update ID_i's column vector $\underline{c}_k^i$ in $\mathcal{KRL}^i$. The update is done by using the majority vote for each element in the column vector, i.e. the majority for each accusation value $a_{l,k}^j$with $l \in \{1, \dots, M_i\}$ is computed. For simplicity, we assume ID_i saved d_k column vectors $\underline{c}_k^j$ from d_k neighbors $ID_j$with $j \in \{1, \dots, d_k\}$ in step 4, with $d_k \geq \varepsilon$. Now ID_i computes

$$a_{l,k}^i = \begin{cases} 1 & \text{if} \sum_{j=1}^{d_k} a_{l,k}^j > \frac{d_k}{2}, \quad \text{with } l \in \{1, \dots, M_i\} \text{ and } j \in \{1, \dots, d_k\} \\ 0 & \text{if} \sum_{j=1}^{d_k} a_{l,k}^j < \frac{d_k}{2}, \quad \text{with } l \in \{1, \dots, M_i\} \text{ and } j \in \{1, \dots, d_k\} \\ a_{l,k}^i & \text{otherwise} \end{cases}$$

Again, only values $a_{l,k}^j$ for nodes ID_l that are within ID_i's m-hop range are considered, others are discarded. If no majority can be found, the accusation value in $\mathcal{KRL}^i$ remains unchanged. Node ID_i repeats this for all column vectors $\underline{c}_k^i$ for which the number of collected vectors $\underline{c}_k^j$ is $> \varepsilon$.

3.2 Example for $\mathcal{KRL}$ update

We present an artificially small and simple network scenario to illustrate how Alg.3.3 works. In our example we consider six network nodes ID_i with $i \in \{1, \dots, 6\}$ as shown in Fig. 1, where the nodes maintain key revocation lists for nodes in two hop distance, i.e. $m = 2$, and the security parameters are set to $\delta = 3$ and $\varepsilon = 2$. We now show how ID_1 updates its revocation list $\mathcal{KRL}^1$ upon receiving the accusation messages am_2, am_3, am_4 from its one-hop neighbors $\mathcal{N}_1 = \{ID_1, ID_2, ID_3, ID_4\}$. To do so, ID_1 executes Alg.3.3. For simplicity, we assume that the current expiry date is t for all nodes and the version number is $v = 1$ for all public keys. Hence, we neglect the values (t, v) in the revocation lists. The revocation lists from ID_3 and ID_4, and ID_1's list from before the update look as follows:

$$\mathcal{KRL}^1 = \begin{pmatrix} ID_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ ID_2 & 1 & 1 & 0 & 0 & 0 & - \\ ID_3 & 0 & 0 & 0 & 0 & 0 & 1 \\ ID_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ ID_5 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathcal{KRL}^3 = \begin{pmatrix} ID_1 & 0 & 0 & 1 & 0 & 0 & 1 & - \\ ID_2 & 1 & 1 & 0 & 1 & 1 & - & - \\ ID_3 & 0 & 0 & 1 & 0 & 0 & 0 & - \\ ID_4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ ID_5 & 1 & 0 & - & 1 & 1 & 0 & 1 \\ ID_6 & 1 & - & - & 1 & 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{KRL}^4 = \begin{pmatrix} ID_1 & 0 & 0 & 1 & 0 & 0 & 0 & - \\ ID_2 & 1 & 1 & 0 & 1 & 1 & - & - \\ ID_3 & 0 & 0 & 1 & 0 & 0 & 1 & - \\ ID_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ ID_5 & 1 & 0 & - & 1 & 1 & 0 & 1 \\ ID_6 & 1 & - & - & 1 & 1 & 1 & 0 \end{pmatrix}.$$

We now go through the steps of Alg.3.3:

(Step 1): save am_3 and am_4, discard am_2 because $R_2^1 = 1$ in $\mathcal{KRL}^1$

(Step 2): am_3 and am_4 successfully authenticated using $K_{1,3}$ and $K_{1,4}$, resp.

(Step 3): use column vectors $\underline{c}_3^3$ from $\mathcal{KRL}^3$ and column vector $\underline{c}_4^4$ from $\mathcal{KRL}^4$ to update column vector $\underline{c}_3^1$ and $\underline{c}_4^1$, respectively , i.e.

$$\underline{c}_3^1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \underline{c}_4^1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

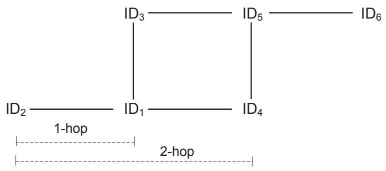

Fig. 1. Example of network setting

(Step 4): from $\mathcal{KRL}^3$ discard $\underline{c}_1^3$ because $k = i = 1$, $\underline{c}_2^3$ because $R_2^1 = 1$, $\underline{c}_3^3$ because $k = j = 3$, $\underline{c}_4^3$ because am_4 was accepted in Step 2, and $\underline{c}_6^3$ because ID_6 is more than 2 hops away. Hence, only save $\underline{c}_5^3$ from $\mathcal{KRL}^3$. For similar arguments, save only $\underline{c}_5^4$ from $\mathcal{KRL}^4$.
(Step 5): since the number of saved columns for the same node ID_5 equals the required minimum, i.e. $d_5 = 2 \geq \varepsilon = 2$, continue with majority vote. The two saved column vectors $\underline{c}_5^3$ and $\underline{c}_5^4$ for ID_5 are used to update the corresponding column vector $\underline{c}_5^1$ and complete the update of $\mathcal{KRL}^1$, i.e.

$$\underline{c}_5^3 = \begin{pmatrix} 1 \\ - \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \underline{c}_5^4 = \begin{pmatrix} 0 \\ - \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \underline{c}_5^1 = \begin{pmatrix} 0 \\ - \\ 1 \\ 1 \\ 0 \end{pmatrix}, \mathcal{KRL}^1 = \begin{pmatrix} ID_1 \; 0\;0\;0\;0\;0\;0 \\ ID_2 \; 1\;1\;0\;1\;1\;- \\ ID_3 \; 0\;0\;0\;0\;0\;1 \\ ID_4 \; 0\;0\;0\;0\;0\;1 \\ ID_5 \; 0\;0\;0\;1\;1\;0 \end{pmatrix}.$$

3.3 Key Renewal

The presented IBC revocation scheme for MANETs needs to be complemented by a key renewal algorithm to enable a node ID_i to obtain a new key pair (Q_i, d_i) after its public key expired, or was revoked by a harakiri message or δ accusation messages. In any case, a node needs to access the off-line KGC for key renewal. When doing so, the node must re-authenticate itself to the KGC using some credentials that identify the node. An off-line KGC cannot distinguish between malicious nodes whose keys have been revoked because of bad behavior or honest nodes whose keys have been compromised. Therefore, malicious nodes can always request new keys once their old keys have been revoked due to malicious behavior. Note that malicious nodes are acting under their true identities and thus 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.

Upon receiving a new key pair and re-joining the network, a node ID_i only needs to broadcast its new public key Q_i' to the m-hop neighborhood, if ID_i received new keys with a version number $v_i' > 1$. The receivers of Q_i', update the version number in their revocation lists accordingly and set all accusation values for Q_i' to zero. In all other cases, the node does not need to inform other nodes about its new keys. This a based on the fact that at each new expiry interval t', all new public keys Q' are assumed to have $v' = 1$ and the accusation values of these new keys are all set to zero, until new accusations are received for Q'.

4 Security Analysis

We assume that the underlying IBC scheme including the pre-shared keys from (1) are secure [1, 2] and limit our analysis to the introduced key revocation and key renewal schemes. In the revocation scheme, trust is based on monitoring one-hop neighbors. Node ID_i who trusts a neighbor ID_j, also trusts that this neighbor properly observes its own neighbors and maintains a correct revocation list. The security of the key revocation scheme depends on security parameter δ, which is the threshold for key revocations. Hence, the system is secure for up to δ-1 colluding malicious nodes. Note that these malicious nodes need to remain undetected in the neighborhood scheme for a successful attack. The security parameter ε protects honest nodes from false accusations that were made by nodes that are more than one but maximally m hops away. In the proposed revocation scheme, nodes do not trust nodes that are not in their direct communication range and a majority vote is used for accusations from these distant nodes. Only if at least $d = \frac{\varepsilon}{2}$ different sources of an accusation agree in their values, an accusation by a distant node is accepted. We would like to point out that the majority vote is computed for each accusation value separately. Consequently, a group of at least $\frac{\varepsilon}{2}$ colluding nodes can manipulate one accusation value for a node ID_i. However, δ such manipulations are necessary to revoke ID_i's key. Hence, the majority vote with parameter ε significantly reduces the propagation of false accusations.

We now briefly discuss some effects of some typical attacks in MANETs on our scheme. A more detailed analysis is in the full version of this paper. We believe that node compromise and selfish nodes are very likely in MANETs. Both can be detected in our neighborhood watch scheme. In our scheme, keys from such malicious nodes are first locally revoked by one-hop neighbors and eventually revoked by all nodes in m-hop distance. In that way malicious users who control compromised nodes are excluded from the network, because they cannot request new keys since this requires authentication to the KGC. On the other hand, selfish nodes are encouraged to participate, because otherwise they are forced to frequently renew their keys which imposes even more costs than forwarding other nodes' messages. Malicious nodes cannot simply drop or manipulate accusations against themselves, because first would be detected by the neighborhood watch scheme, and second attempt would be prevented by using majority votes for accusations. A roaming adversary ID_i may move every time its number of accusation approaches δ. In another scenario, a roaming adversaries moves to a new neighborhood more than m hops away, such that nobody has any accusation values for ID_i. However, in both scenarios with roaming adversaries, the present one-hop neighbors quickly detect ID_i's malicious behavior and locally revoke its key. With those accusations propagating to all m-hop neighbors the adversary would need to move faster than the propagation of the accusations. Hence, the power of roaming adversary for launching attack is fairly limited since they need to move fairly fast and frequently and cannot remain at the same location for a longer period of time. Malicious nodes may try to bypass the security parameter δ by fabricating δ different identities. However, this is prevented by the KGC which checks the identity of every node before issuing keys, where the security of the scheme is based on the honesty of the KGC.

5 Related Work

5.1 IBC schemes in MANETs

Recently, two IBC schemes have been introduced for securing MANETs [4, 8]. Both papers suggest emulating a distributed on-line KGC using (k, n)-threshold schemes. As mentioned earlier, the use of threshold schemes introduces a significant communication and computational overhead to the network. 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 due to the efficient key management of the underlying IBC schemes. However, both solutions do not introduce key revocation and key renewing mechanisms for their schemes.

Our proposed key revocation and renewal schemes can be applied to both IBC schemes in [4, 8] to provide key revocation and key renewal in MANETs. Our schemes can be easily modified for distributed on-line KGCs, such that revocation is done by all network nodes by executing Alg.1-3 from Sect. 3.1, where the distributed on-line KGC takes over the task of key renewal using the key format from (2). Since our revocation scheme works independent of the (k, n)-threshold schemes, the solution is very efficient.

5.2 PKI schemes for MANETs

Many PKI-based schemes for MANETs have been introduced, e.g. [3, 9, 11], where some of them use (k, n)-threshold schemes to implement distributed on-line CAs [9, 11]. In [11], it is suggested to collaboratively revoke certificates, but no algorithm is introduced. In fact, a revocation scheme in this solution would require threshold signatures, which is computationally very demanding. In [9], an accusation scheme is briefly outlined, in which each node observes their neighboring nodes for malicious behavior. Based on their observations, nodes send their signed accusations to an m-hop neighborhood. All receivers verify the accusations and update their accusation lists accordingly. If the number of accusations for one node is greater than a threshold δ, the certificate is revoked. Compared to [9], our revocation scheme uses pre-shared keys to secure accusation messages instead of signatures. Hence, our scheme is more efficient, once the pre-shared keys have been computed for the first time. Furthermore, the problem of newly joining nodes is not discussed in [9]. In our solution, new nodes can simply start the revocation scheme (Alg.1-3), whereas in [9] joining nodes need to verify accusation tables from its neighbors to learn about past accusations and revocations. This requires the verification of all received accusation values, i.e. approximately N^2 verifications for N network nodes which is clearly too demanding. Furthermore, a harakiri algorithm for nodes that want to revoke their own keys is not discussed in [9].

In [3], a certificate revocation scheme for MANETs is presented that uses an accusation scheme with threshold δ. The scheme assumes an off-line CA that issues certificates to all network nodes before they join the network. All accusations are frequently broadcasted throughout the entire network. The revocation scheme uses a weighted accusation scheme to decide wether a certificate is revoked. Here, instead of just computing the sum of accusations, the accusations are weighted according to the number of accusations a node made, how many accusations were reported against this node, etc.. When a new node joins the network, the node receives the accusation tables from all network nodes. The accusation messages in [3] are not secured at all and the authors suggest to check for inconsistencies in received accusation tables and only trust accusations from senders with sufficient trust value. Compared to [3], our scheme secures accusation messages and we provide a detailed description how majority votes can be implemented to check for inconsistencies in accusation values. Furthermore, our scheme provides scalable performance by choosing an m-hop propagation range for accusations. A harakiri algorithm for nodes that want to revoke their own keys is not discussed in [3]. Note that the weighted accusation scheme from [3] can be applied to our revocation scheme, such that weighted accusation values are used in Alg.3.1.

6 Discussions and Conclusions

In this paper, we introduced the first key revocation and key renewal schemes for IBC schemes in MANETs. The proposed key revocation and key renewal schemes can be applied to the recently proposed IBC schemes for MANETs [4, 8] which do not provide these fundamental mechanisms. Furthermore, our solution is applicable to any kind of IBC scheme in MANETs and can be easily adapted to PKI schemes in MANETs with off-line or on-line CAs. Our neighborhood watch scheme helps to detect malicious, selfish, and any other misbehaving nodes in MANETs, where all observations are securely propagated to an m-hop neighborhood.

The proposed revocation scheme is scalable in its security and performance parameters m, δ, and ε. For instance, greater m decreases the chances of a roaming adversary to remain undetected, where smaller values increase the scheme's performance with respect to bandwidth and memory space. Security parameters δ and ε prevent up to δ-1 undetected colluding one-hop neighbors and at least that many m-hop neighbors from falsely revoking keys. Our solution is very efficient due to the use of pre-shared keys to secure accusation messages instead of signatures and propagating messages to an m-hop neighborhood instead of to the entire network. Other than existing PKI solutions for MANETs, our solution provides a very efficient way for nodes to revoke their own keys. Furthermore, newly joining nodes can simply join the network and start the revocation scheme without first verifying a large number of past accusations and revocations.

References

1. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing, *Advances in Cryptology - CRYPTO '2001*, LNCS 2139, pp. 213-229, 2001.
2. 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.
3. 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.
4. 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.
5. 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.
6. K. Hoeper and G. Gong. Identity-Based Key Exchange Protocols for Ad Hoc Networks, *Canadian Workshop on Information Theory –CWIT'05*, 2005.
7. D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, vol. 353, chapter 5, pp. 153–181. Kluwer Academic Publishers, 1996.
8. 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.
9. 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.
10. C.E. Perkins, E.M. Royer, and S.R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF Internet draft, Internet Draft (draft-ietf-manet-aodv-09.txt), November 2001. Work in Progress.
11. L. Zhou and Z.J. Haas. Securing Ad Hoc Networks, *IEEE Network Journal*, vol. 13, no. 6, 1999, pp. 24-30.