SECRET: A Secure and Efficient Certificate Revocation Scheme for Mobile Ad Hoc Networks

Dieynaba Mall[1], Karim Konaté[1], and Al-Sakib Khan Pathan[2]
[1]Department of Mathematics and Computer Science, Université Cheikh Anta Diop de Dakar, Dakar, Senegal
[2]Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia
dieynaba.mall@ucad.edu.sn, karim.konate@ucad.edu.sn, sakib@iium.edu.my

***Abstract*—The intent of this paper is to propose an enhanced certificate revocation scheme for Mobile Ad hoc Networks (MANETs). Our approach is built on mainly two previously proposed mechanisms. A combination of the schemes and optimization of certain steps with intelligent choices of parameters could significantly reduce the overhead associated with such mechanism. We prove the efficiency of our approach by performance analysis. Also, we present the security analysis that shows clear gains than the previously proposed schemes.**

Keywords- ad hoc; certificate; key; mobile; network; revocation

I. INTRODUCTION

In Mobile Ad hoc Networks (MANETs), the nodes maintain the network by communicating among themselves without any particular centralized entity. The ad hoc environment raises critical security challenges. The wireless mode of communication and dynamic nature of MANETs make them susceptible to more security attacks than those of the traditional wired networks. In such environment, eavesdropping, message injection, Denial of Service (DoS) and battery exhaustion attacks are relatively easy to launch. Moreover, router (or, a node that acts as the forwarder of packets) and terminal nodes are often not reliable.

In order to protect routing information and data for application, often cryptographic techniques are essential. Cryptographic keys are used as evidence of reliability to authenticate a node as a legitimate member of the network. They are also used to ensure confidentiality and integrity. A secure and effective cryptographic key management is crucial for a reliable network service and thus, for the success of wireless ad hoc networks. The major challenges of implementing PKI (Public Key Infrastructure) in MANETs are: issuing, distributing certificates and enabling certificate revocation.

There are mainly two PKI implementations in MANETs: one with a Certification Authority (CA) and another with a distributed on-line CA. In the first case, the CA issues the certificates to nodes before they join the network. We call this an *off-line CA*. In the second case, the nodes obtain their certificates from a group of network nodes that act as distributed *on-line CA*.

Many PKI-based schemes have been proposed to secure MANETs and similar resource constrained networks [1], [2], [3], [4], yet many of them do not provide certificate revocation mechanisms at all. In this paper, our objective is to improve some PKI-based key management solutions in MANETs. We present a key revocation scheme based on a solution that has been already proposed for identity-based schemes.

This paper is organized as follows: After, Section I, Section II reviews existing solutions related to our work. Section III presents our motivation. This section is followed by an overview of the schemes that serve as the building blocks of our proposal in Section IV. Section V describes our key revocation scheme; Section VI discusses the security and performance analysis, and finally, Section VII concludes this paper.

II. RELATED WORKS

The mechanism how a key can be revoked from a network is a critical part of security management [5]. However, despite its importance, many PKI and Identity-Based Cryptographic (IBC) solutions for MANETs ignore key revocation or just briefly outline an idea for a solution without providing actual algorithms [1-2], [6-8], [21]. According to our investigation in this area, so far only a few practical and comprehensive revocation schemes for MANETs have been proposed [9-14].

In [9], Luo et al. briefly describe a localized certificate revocation scheme based on (t, N) threshold mechanism to distribute the capabilities of the CA to all the network nodes. In their proposal, each node monitors its neighboring nodes and disseminates its signed accusation to its *m*-hop neighborhood. Upon receiving this accusation, each node would first verify the trustworthiness of the accuser before updating its TRL (Ticket Revocation List) accordingly. Each entry of the TRL contains a node ID and a list of the node's accusation from others. The certificate of a node would be revoked when the number of accusations against this node exceeds a predefined revocation threshold.

In [10], Arboit et al. present the first self-organized certificate revocation scheme for MANETs. Their revocation protocol is based on weighted accusations. The weight is determined from a node's reliability which is derived from its past behavior. More specifically, the weight of a node's accusations depends on the number of accusations against it as well as the number of additional accusations made by it. All accusations are frequently disseminated throughout the entire network. In order to avoid the disadvantages related to the use of digital signature, the authors have suggested using the one-way hash chains to provide the authentication of data origins and the integrity check of messages.

Instead of revoking keys, in [13], the authors propose to reelect the nodes that wish to obtain a new key by a group of nodes. This reelection is based on the observed behavior, and

on secret key sharing. To avoid costly secret key reconstructions, the authors also outline a lightweight scheme in which nodes periodically broadcast a so-called buddy-list with identifiers of trusted nodes. A third, very radical approach in [13] is to revoke the keys of accusing and accused nodes.

In [14], a distributed on-line Key Generator Center (KGC) consisting of *n* network nodes (called D-KGC) carries out key revocations and renewals. The distributed KGC is implemented using a *(k, n)*-threshold scheme. The nodes monitor their neighborhood and send their accusations to *b* assigned D-KGCs. Once a threshold is reached, a group of *k* DKGCs collaboratively sign a revocation message.

In [11], a node predicts the behavior of the other nodes based on its own observations and reports from neighbors, by using a 3-dimensional Dirichlet distribution. To illustrate a bit, a Dirichlet distribution is a multivariate generalization of a beta distribution which can be conceptualized as a probability distribution of PMFs (Probability Mass Functions) [22]. In this approach as presented in [11], the nodes can have three different states: *trustworthy*, *suspicious*, and *malicious*, which would enable multilevel responses.

In [12] each node uses a neighborhood watch scheme to monitor the nodes within its 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 accuse this one. This key revocation scheme is scalable in parameters *m* and δ, i.e., the level of security can be chosen as performance tradeoff. This scheme is further analyzed in [15] and extended in [16].

III. Motivation for Our Work

By minimizing the importance of the authentication issue in MANETs, most of the proposed certificate revocation schemes in MANET present many insufficiencies: they are vulnerable to various types of attacks and they do not provide a way to efficiently exploit the limited resources of nodes in such networks. Only those schemes that are based on digital signatures have addressed this issue. However, the cost associated with using such operations is still substantially higher than that of symmetric cryptographic operations; because if they are frequently performed, they substantially consume the battery power of nodes. An attacker can inject bogus broadcast packets to force the network nodes to execute expensive signature verification. This renders the signature-based broadcast authentication protocols easily vulnerable to DoS attacks.

In order to minimize the use of digital signature for the purpose of broadcast authentication and efficiently exploit the limited resources of MANET, [10] suggested using one-way hash chains. Although this scheme reduces the resource consumption, it does not handle the vulnerability against DoS attacks because it does not allow immediate authentication. In other words, nodes have to forward the received broadcast packet before properly authenticating it due to the delayed key disclosure.

In order to address the two main issues mentioned above (i.e., vulnerability against various attacks and resources consumption), we propose in this paper a certificate revocation scheme that is based on a solution that was previously proposed for identity-based schemes in MANETs. Our proposal is to adapt the proposal in [16] to PKI systems by employing the HEAP protocol [17] as the underlying broadcast authentication scheme.

IV. Building Blocks of Our Proposal

A. Rationale for the choice of [16] and HEAP

Among all the proposed revocation solutions, the proposal in [16] seems to be the most efficient that takes into account most of the flaws or vulnerabilities of the other existing solutions, because it gives solutions to the problems related to / to the:

- securing of accusation messages ;
- offering of resistance against several types of attacks such as: attacks by spoofing identity, attacks by coalition of nodes, the Sybil attacks, attacks of roaming opponents, DoS attacks, etc;
- inaccuracies inherent in the monitoring system implemented by the nodes ;
- verification of past accusations by newly joining nodes ;
- computational and communication costs.

Compared to the other existing authentication protocols (e.g., Tesla [18], LHAP [19], [20]), HEAP has been proven to be the most secure and efficient scheme. Besides that, it provides the means that could be used to fulfill the requirements of our design goal.

B. Overview of the proposal in [16]

In [16], Hoeper and Gong present a self-organized key revocation scheme for MANETs. The authors introduce a new type of scheme involving node's identity, key expiry date, and key version number for identity-based public keys such that new keys can be issued for the same identity after the previous key has been revoked. In their revocation scheme, each node uses a neighborhood watch mechanism to monitor the nodes within its communication range. Upon detection of malicious behavior, these observations are then securely propagated to an *m*-hop neighborhood (*m* denotes the range of the accusation propagation) using the pre-shared secret key obtained from a non-interactive identity-based key agreement protocol. Furthermore, when a node realizes that its private key has been compromised, it generates a *harakiri* message and propagates it to the entire network. Node *A* would consider node *B*'s public key as revoked if at least one of the following three conditions is true: (i) *A* observes *B*'s malicious behavior through the neighborhood watch; (ii) A receives a *harakiri* message from *B* declaring that its private key has been compromised; (iii) *A* receives at least δ accusations against *B* from trustworthy nodes within node *A*'s *m*-hop neighborhood, where δ is a pre-determined revocation threshold. The authors also use the majority vote with a parameter ε to mitigate the influence of false accusation attacks from colluding *l*-hop neighbors $(2 < l < m)$. Moreover, the newly joining nodes can simply join the network and start the key revocation scheme without first verifying a large number of past accusations. Four

algorithms associated with the basic revocation scheme are: *Algorithm 1: Neighborhood watch*, *Algorithm 2: Harakiri*, *Algorithm 3: Propagate*, and *Algorithm 4: Update KRL (Key Revocation List)*. Figure 1 gives an overview of the scheme.

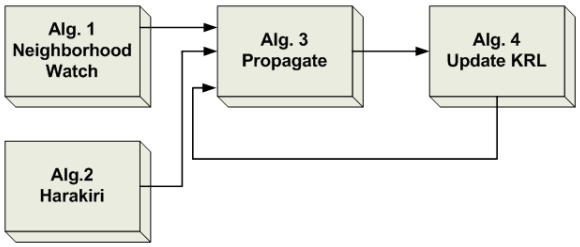

Figure 1. Overview of the revocation scheme in [16].

The revocation mechanism described in [16] for identity based schemes is efficient because it uses pre-shared keys to secure accusation messages instead of signatures and the propagation of accusation messages from neighborhood watch or update of *KRL* to an *m*-hop neighborhood instead of to the entire network. However, it cannot be directly applied to PKI-based schemes. That is mainly due to two factors in particular:

- the use of self-authenticated public keys in identity based protocols;
- the use of pre-shared keys to secure the accusation messages.

Indeed, in identity-based protocols, the users' public keys and identities do not need to depend on certificates. Thus, public keys in identity-based protocols are self-authenticated, which is not the case in the other types of protocols where a public key's authenticity requires to be established. Once this authenticity is established thanks to certificates, it is no longer possible to revoke keys without using the revocation of certificates. Moreover, the proposed key revocation mechanism for identity-based protocols assumes the existence of pre-shared keys to secure accusation messages. Each pair of nodes is able to calculate a pairwise pre-shared secret key in a non-interactive way [16] as described in eqn. (1).

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

where, d_i , d_j are the private keys, Q_i, Q_j are the public keys of nodes *i* and *j* respectively, and ê is a bilinear mapping.

The use of pre-shared secret keys is especially supported by the possibility of calculating them in a non-interactive way. This makes it possible to reduce the total computational and network overhead.

C. Overview of HEAP protocol

In the bootstrapping phase of HEAP [17], a new node *i* first generates a single key called *ikey* and one pairwise key for each one-hop neighbor *j*, called *okey*. The keys are fixed sized random bit strings and are cheap to generate. Then, node *i* uses a standard key exchange mechanism and PKI using trusted third parties to securely share the group key *ikey* with the corresponding neighbors. Whenever a node's neighborhood changes due to mobility, it shares a new group key and pairwise key with each new neighbor. After the key generation and distribution phase, packet authentication can take place; this phase can be described as follows:

When a node *i* wants to broadcast a message to all of its one-hop neighbors, by enabling each of them to authenticate the origin of the packet, it can proceed by generating a new *HMAC* (Hash based Message Authentication Code) [23] for each neighbor *j* using its $okey_j$. However, this would be computationally too expensive (see Figure 2).

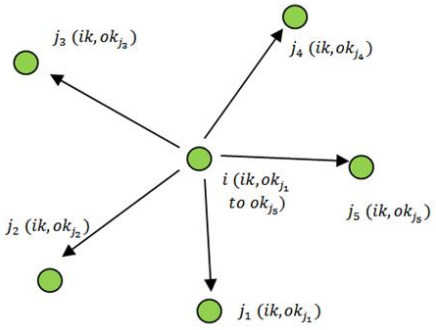

Figure 2. Node i needs to transmit a packet to its neighbors j_1 through j_5.

To reduce the computational cost, the authors have proposed a more efficient algorithm using a slightly modified HMAC. The original HMAC is computed by:

$$HMAC(M,K) = H(K\ XOR\ opad \mid H(K\ XOR\ ipad \mid M)) \quad (2)$$

where *H(x)* is a hash function such as MD5 or SHA-1, *M* is the message to be transmitted, '*opad*' is the hexadecimal number 5C used to pad each byte of *K* up to one block size. '*ipad*' is the hexadecimal number 36 used to pad each byte of *K* up to one block size. The symbol '|' represents concatenation.

The computation of HMAC in eqn. (2) can be done as:

Step 1: $H(K\ XOR\ ipad \mid M)$

Step 2: $H(K\ XOR\ opad \mid Hash\ from\ step\ 1)$

By doing so, it can be seen that step 1 will result most of the computational overhead. Thus, for the modified algorithm, the authors propose to use two keys $ikey$ and $okey$. The first key *ikey* is padded with 0's to make it one block size before being used to generate the hash in *New Step1*: $H(ikey \mid M)$

Note that the sender needs to compute this expensive step only once because all the one-hop neighbors share the same key *ikey*. To prevent the impersonation of node *i* by anyone of its one-hop neighbors, the pairwise key *okey* is used to generate the hash in new step 2. $okey_j$ is padded with 0's to make one block before being used . *New Step 2*: $H(okey_j \mid Hash\ from\ step\ 1) = MAC_j$. This step needs to be computed once for each one-hop neighbor j.

To protect against message replays, index numbers are included in the packet before transmission. The index number is concatenated with the payload data to give the message *M*. The index is incremented by one for each subsequent message. The format of each packet sent by a node *i* is then:

$i \rightarrow *: M, index, Mac_1(M \mid index), \ldots, MAC_n(M \mid index)$

Upon receiving this message, a one-hop neighbor node *j* computes the MAC to see if it matches any of the MAC tags in the message and whether the index number is valid. If it is the case, the message is accepted as authentic; otherwise, it will be dropped.

V. Our Revocation Scheme

A. System Assumptions

For the proposed revocation protocol, we assume a PKI-based system with an external trusted certificate authority, CA. We consider that each node can communicate with this trusted CA before joining the network and can obtain a unique public key certificate signed by the CA as well as the authentic public key of the CA.

TABLE I. List of Notations for Certificate Revocation Scheme

N	total number of network nodes
$N_{1,i}$, σ_i	set and number of *i* 's one-hop neighbors
N_i, Ω_i	node *i*'s perception of set and number of network nodes
δ, ε	thresholds for revocation and reported accusation
α, β	false positive and false negative rates of monitoring scheme
m	propagation range of accusation messages

We also assume that all direct communication links between nodes are bidirectional and each node has an implemented monitoring scheme. These two assumptions are necessary to enable nodes to monitor their neighbors in a communication range. We assume finally that each node knows its one-hop neighbors - this is necessary to assure a complete distribution of shared keys *ikey* and $okey_j$ mentioned in subsection IV. C.

B. Description of the proposed mechanism

In this section, we present a modified version of the revocation scheme described in [16]. Our proposal comprises of three algorithms which work similarly like those described in [16]. Table 1 shows the basic notations used to describe our scheme.

Certificate Revocation Lists (CRLs) - Each node i creates a certificate revocation list CRL^i for any of its known nodes *j* such that $j \in N_i$. This list can be represented by a matrix with dimensions $(\Omega_i, \Omega_i + 3)$ as described below:

$$CRL^i = \begin{pmatrix} a^i_{1,1} & \cdots & a^i_{\Omega_i,1} & cert_1 & vp_1 & X^i_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a^i_{\Omega_i,1} & \cdots & a^i_{\Omega_i,\Omega_i} & cert_{\Omega_i} & vp_{\Omega_i} & X^i_{\Omega_i} \end{pmatrix}$$

Each j-th column vector in CRL^i for $1 \leq j \leq \Omega_i$, contains all accusations $a^i_{k,j}$ made by node *j* against nodes $k \in N_i$. Each j-th row vector in CRL^i for $1 \leq j \leq \Omega_i$, corresponds to a node $j \in N_i$ and contains among other information, the accusation values $a^i_{j,k}$ from all nodes $k \in N_i$ evaluating node j. Element $(\Omega_i + 1)$ in each j-row contains the serial number $cert_j$ of node j 's certificate, element $(\Omega_i + 2)$ represents the validity period vp_j of node j's certificate, and element $(\Omega_i + 3)$ contains a 1-bit flag X^i_j that, when set, indicates that node i considers certificate of node j as revoked.

Revocation Scheme – Here, we use: (i) certificate revocation lists instead of key revocation lists, and (ii) the HEAP protocol as broadcast authentication scheme. Hence, shared keys *ikey* and $okey_j$ are used to secure accusation messages. The combination gives significant advantage over the previous approach.

Algorithm 1: *Neighborhood Watch*

In this algorithm each node i monitors its one-hop neighbors. Whenever it observes a suspicious neighbor $j \in$, $N_{1,i}$, it sets $a^i_{j,i} = 1$ and creates a neighborhood watch message nw_i with:

$$nw_i = M, index, MAC_1, MAC_2, \ldots, MAC_{\sigma_i}$$

where,

- $M = cert_i, CRL^i, hopcount$, containing ;
  - $cert_i$ is the serial number of *i*'s certificate;
  - $hopcount$ ensures that the message reaches all nodes in *m*-hop distance. Initially, node i sets $hopcount = m$.
- $index$ is the index number related to this message and used to prevent replay attacks.
- MAC_1 , MAC_2 , … , and MAC_{σ_i} are the different $MACs$ computed each for a one-hop neighbor $j \in N_{1,i}$ according to the *New Step 2* described in subsection IV.C.

After computing this neighborhood watch message, node i starts Algorithm 2 to propagate it.

Algorithm 2: *Propagate*

This algorithm is triggered by Algorithms 1 and 3. After creating an accusation message which can be neighborhood watch message nw_i or update message um_i , the nodes securely propagate accusations to their one-hop neighbors.

Algorithm 3: *Update CRL*

This algorithm describes how the node i updates its own revocation list CRL^i according to the received accusation message. Note that i prepares an update message um_i for all its one-hop neighbors $j \in N_{1,i}$ with:

$$um_i = M, index, MAC_1, MAC_2, \ldots, MAC_{\sigma_i}$$

where:

- $M = cert_i, CRL^i, hopcount$, containing ;
  - $cert_i$ is the serial number of *i*'s certificate;
  - $hopcount$ ensures that the message reaches all nodes in m-hop distance. Initially, node i sets $hopcount = m$.

- $index$ is the index number related to this message and used to prevent replay attacks
- MAC_1 , MAC_2 , … , and MAC_{σ_i} are the different MAC s computed each for a one-hop neighbor $j \in N_{1,i}$ according to the *New Step 2* described in subsection IV.C.

For more details about these algorithms, the interested reader is referred to [16].

VI. Security and Performance Analysis

A. Security Analysis

Detection and thwarting attacks from insiders are the subjects of Intrusion Detection and Response Systems (IDRS). However, to guarantee the correctness of such systems, the systems must be established on a secure and efficient authentication protocol that could provide a high level of protection against attacks from outsiders. The use of HEAP as authentication scheme provides a foundation which our response system can be based on to efficiently cut-off a compromised insider from the network. With HEAP, our revocation scheme can authenticate every single packet in every single hop. Hence, it can combat various attacks from outsiders such as: replay, impersonation, DoS, man-in-the-middle, wormhole attacks, etc. In addition, HEAP offers some level of protection against insider attackers who try to forge packets and impersonate other insiders.

Our proposal considers a wide range of insider attacks and proposes the use of some intelligent techniques, security parameters, δ, ε, system parameters α, β, and parameter m to protect the system. However, malicious nodes may try to bypass these parameters by:

- Launching Sybil attacks by fabricating δ different identities or by creating δ rows in its CRL for δ non-existing nodes and making them to accuse the same node in its CRL. However, in our scheme, a node can only get one valid certificate and one valid public key for a specific period of validity. Thus, the first described kind of Sybil attack is not applicable because that requires keying material. The second type of Sybil attack is prevented in our scheme by the second revocation threshold, ε which is fixed for reported accusations.
- Dropping accusations against themselves. In such a case, the malicious node's chance in succeeding is very limited because: (i) even if one of the propagation paths is broken, accusations still would reach other nodes, (ii) it incurs the risk of being detected by the *neighborhood watch* scheme.
- Attempting to modify accusations against themselves. In our proposal, a malicious node would be discouraged simply because each accusation's integrity is guaranteed using some one-hop shared keys.
- Moving to a new neighborhood whenever its accusation account approaches δ. This threat can be thwarted by adjusting the parameter m and the period of validity of certificates accordingly, and by encrypting all accusations.

Finally, to counteract attacks by colluding nodes, we can choose the security parameters δ, ε and the system parameters α, β in accordance with the bounds and relations derived in [16].

B. Performance Analysis

Due to the resource constraints in MANETs, we propose to evaluate the performance of our proposal in terms of storage, computational, and communication overhead.

We assume a fairly static network in which all nodes have exchange shared keys with their one-hop neighbors.

In our scheme, to participate in the revocation process of a malicious node, each node must store $2\sigma_i$ keys, σ_i certificates, and σ_i index numbers. To send a neighborhood watch message nw_i or an update message, um_i, node i will have to compute one MAC using the group key $ikey$ and $(\sigma_i - 1)$ MACs using the different $okey_j$. Each node j receiving this accusation message will compute one MAC using its key $okey_j$. Thus, in our protocol, the computational overhead related to an accusation is equal to: $(2\sigma_i - 1)$ MACs.

In our scheme, a neighborhood watch message nw_i or an update message. um_i , contains $(\sigma_i - 1)$ hash values, one CRL , one certificate's serial number $cert_i$, one index number, and one $hopcount$. This message is transmitted only once by broadcast.

To evaluate these results, we also evaluate the storage, computational and communication overhead generated by an accusation message in [16]. Here also, we assume a fairly static network in which after the initial phase, all pre-shared keys are computed, i.e., for all senders as well as receivers, each of them computes its pairwise pre-shared key $k_{i,j}$ according to eqn. (1).

TABLE II. Performance Comparison

	Our Approach	***The Approach in [16]***
Storage overhead	$2\sigma_i$ *keys* $+ \sigma_i$ *certificates* $+ \sigma_i$ *index numbers*	σ_i *ID* $+ \sigma_i$ *public keys* $+(\sigma_i - 1)$ *shared keys*
Computational overhead (nw_i or um_i)	$(2\sigma_i - 1)$ *MACs*	$(2\sigma_i - 1)$ *MACs*
Communication overhead (nw_i or um_i)	$(\sigma_i - 1)$ *hash values* $+ CRL + cert_i$ $+$ *index number* $+$ *hopcount*	$(\sigma_i - 1)$ [*hash values* $+$ $KRL + ID +$ *hopcount*]

In [16], to participate in a revocation process, a node needs to store the σ_i identities, σ_i public keys, and $(\sigma_i - 1)$ keys are shared with its one-hop neighbors. In this mechanism, a neighborhood watch message, nw_i or an update message, um_i , will generate $2(\sigma_i - 1)$ MACs because the sender will compute $(\sigma_i - 1)$ MACs for its one-hop neighbors and each receiver will in turn compute one

MAC to authenticate the received message. A neighborhood watch message, nw_i or an update message, um_i in [16] includes one hash value, one identity, one Key Revocation List (KRL), and one $hopcount$. However, this message will only inform one neighbor. Consequently, to inform all its one-hop neighbors, node i has to send $(\sigma_i - 1)$ messages. Table II shows the results of our comparative performance analysis.

As it can be seen from Table II, the performance evaluation of our proposal gives satisfactory results when compared to the proposal in [16]. With our approach, the memory space required to store the required information slightly increases due to the storage of certificate of each one-hop neighbor. However, the computational overhead generated by an accusation message in our solution remains the same as the one associated with the proposal in [16] since exactly $(2\sigma_i - 1)$ $MACs$ operations are executed as in [16]. In our scheme, to disseminate an accusation message to the one-hop neighborhood, a node i just needs to execute one broadcast. Thus, compared to the method in [16], our solution considerably reduces the communication overhead associated to the propagation of accusations. Note that in [16], due to the use of pairwise pre-shared secret keys $k_{i,j}$, to propagate an accusation, it is required to unicast the associated message to each one-hop neighbor.

VII. CONCLUSIONS

In this paper, we propose a certificate revocation mechanism that adapts the proposal in [16] to the PKI-based schemes by using HEAP as broadcast authentication protocol. Security and performance analyses show that our approach ensures good protection against a wide range of attacks launched by outsiders as well as by insiders in a cost effective way since our scheme offers smaller overheads.

ACKNOWLEDGMENT

This work was supported in part by Networking and Distributed Computing (NDC) Laboratory, KICT, IIUM.

REFERENCES

[1] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," IEEE Network Journal, Vol. 13, No. 6, 1999, pp. 24-30.

[2] 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, pp. 567-574.

[3] C. Crépeau and C.R. Davis, "A Certificate Revocation Scheme for Wireless Ad Hoc Networks," Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03), 2003, pp. 54-61.

[4] A.-S. K. Pathan and C. S. Hong, "Feasibility of PKC in Resource-Constrained Wireless Sensor Networks," Proc. of the IEEE IDCS'08 in conjunction with 11th IEEE ICCIT'08, December 24, 2008, Khulna, Bangladesh, pp. 13-20.

[5] D. Mall, K. Konaté, and A.-S. K. Pathan, "On the Key Revocation Schemes in Wireless Sensor Networks," The 2013 IEEE International Conference on Green Computing and Communications (GreenCom 2013): Security, Privacy, and Trust Computing, August 20-23, 2013, Beijing, China, pp. 290-297.

[6] H. Deng, A. Mukherjee, and 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, 2004, pp. 107-111.

[7] J. P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," ACM Symposium on Mobile Networking and Computing (MobiHOC 2001), 2001, pp. 146-155.

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

[9] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Roubust Access Control for Mobile Ad Hoc Networks," IEEE/ACM Trans. on Net., Vol. 12, No. 6, 2004, pp. 1049-1063.

[10] G. Arboit, C. Crépeau, C. R. Davis, and M. Maheswaran, "A Localized Certificate Revocation Scheme for Mobile Ad Hoc Networks," Ad Hoc Network, Vol. 6, No. 1, 2008, pp. 17-31.

[11] X. Fan and G. Gong, "Key Revocation based on Dirichlet Multinomial Model for Mobile Ad Hoc Networks," 33rd IEEE LCN 2008, 14-17 Oct. 2008, pp. 958 - 965.

[12] K. Hoeper and G. Gong "Key Revocation for Identity-Based Schemes in Mobile Ad Hoc Networks," International Conference on AD-HOC Networks Wireless (AD HOC NOW'06), LNCS 4104, Springer Verlag, 2006, pp. 224-237.

[13] T. Moore, J. Clulow, S. Nagaraja, and R. Anderson, "New Strategies for Revocation in Ad-Hoc Networks," Security and Privacy in Ad-hoc and Sensor Networks, 4th European Workshop, ESAS 2007, LNCS 4572, 2007, pp. 232-246.

[14] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing Mobile Ad Hoc Networks with Certificateless Public Keys," IEEE Transactions on Dependable and Secure Computing, Vol. 3, No. 4, 2006, pp. 386-399.

[15] K. Hoeper, "Authentication and Key Exchange in Mobile Ad Hoc Networks," Ph.D. thesis, Univ. of Waterloo, Waterloo, Canada, 2007.

[16] K. Hoeper and G. Gong, Monitoring-Based Key Revocation Schemes for Mobile Ad Hoc Networks: Design and Security Analysis. Technical Report 9 2009-15, Centre for Applied Cryptographic Research, March 2009.

[17] R. Akbani, T. Korkmaz, and G. V. S. Raju., "HEAP: hop-by-hop efficient authentication protocol for Mobile Ad-hoc Networks," In Proceedings of the 2007 spring simulation multiconference - Volume 1 (SpringSim '07), Vol. 1. Society for Computer Simulation International, 2007, San Diego, CA, USA, pp. 157-165.

[18] A. Perrig, R. Canetti, D. Song, and D. Tygar, "The TESLA Broadcast Authentication Protocol," In RSA Cryptobytes, 2002, pp. 2-13.

[19] S. Zhu, S. Xu , S. Setia, and S. Jajodia, "LHAP: a lightweight hop-by-hop authentication protocol for ad-hoc networks," 23rd International Conference on Distributed Computing Systems Workshops, 2003. pp. 749 – 755.

[20] B. Lu and Pooch, U.W., "A lightweight authentication protocol for mobile ad hoc networks," International Conference on Information Technology: Coding and Computing, 2005 (ITCC 2005), Vol. 2, 2005. PP. 546 – 551.

[21] K. Hoeper and G. Gong, "Identity-Based Key Exchange Protocols for Ad Hoc Networks," Canadian Workshop on Information Theory (CWIT'05), 2005.

[22] L. Hortensius, "Dirichlet Distribution," available at http://www.tc.umn.edu/~horte005/docs/Dirichletdistribution.pdf [Last accessed: 28 April, 2014]

[23] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication. Internet Request for Comments (RFC 2104), February 1997.