Military Technical College

Kobry El-Kobbah,

Cairo, Egypt

8th International Conference on Electrical Engineering

ICEENG 2012

A Threshold Revocation Scheme For Mobile Ad Hoc Networks

Fatma Elsayed, Hisham Dahshan, Alaa Eldin Rohiem, Ali Elmoghazy

Military Technical College hishamdahshan@yahoo.com,

alaa_rohiem@yahoo.co.uk

***Abstract*—**Security is very important for the reliable operation of mobile Ad Hoc networks (MANETs). One of the critical security issues in MANETs is the revocation of misbehaving nodes. In this paper, we introduce a Threshold Revocation Scheme for Mobile Ad Hoc Networks. In our proposed scheme, the master private key is split into *n* pieces according to a random polynomial. Meanwhile, the master private key could be recovered by combining any threshold *t* pieces based on Lagrange interpolation and hence this master private key is used to sign the revocation message. Because of the decentralized nature of our proposed scheme, it enables a group of legitimate nodes to perform fast revocation of a nearby misbehaving node. Consequently, the proposed scheme improves the safety levels in MANETs. The advantages of the proposed scheme are justified through extensive simulations.

***Index Terms*—**decentralized, revocation, MANET networks.

I.INTRODUCTION

MANET does not rely on a fixed infrastructure for its operation. MANET is an autonomous transitory association of mobile nodes that communicates with each other over wireless links. Nodes that lie within each other's transmission range can communicate directly and are responsible for dynamically discovering each other. In order to enable communication between mobile nodes that are not directly within each other's transmission range, intermediate nodes act as routers, forwarding packets for other mobile nodes in the network that may be multiple hops away from each other. The absence of centralized authority and the infrastructureless nature make MANETs good for emergency, military and fast deployment communications. Security in MANET is an essential component to provide the network with the basic functions such as routing and packet forwarding. Security schemes for MANETs generally employ one or more of the following cryptographic technologies: symmetric-key cryptography, digital certificates or threshold cryptography.

The security goals in MANET include the functionality that is required to provide a secure networking environment. It comprises authentication, access control, confidentiality, integrity, nonrepudiation, and availability. Robust and efficient key management is used to achieving these security goals. Key management is a central part of the security of MANETs. Key management deals with key generation, key storage, distribution, updating, revocation, deleting, archiving, and using keying materials in accordance with security policies.

In MANET, key management can be classified into two types; the first is based on a centralized trusted third party (*TTP*) [1]. The *TTP* is responsible for issuing, revoking, renewing, and providing keying material to nodes. For example, certification authority (*CA*) is the *TTP* in asymmetric cryptosystems; a key distribution center (*KDC*) is the *TTP* in the symmetric system.

The second type of key management is the self-organized key management schemes [2, 3]. Self-organized schemes allow nodes to generate their own keying material, issue public-key certificates to other nodes in the network based on their knowledge. Certificates are stored and distributed by the nodes.

A critical part of any certificate-management scheme is the revocation of misbehaving nodes. Certificate revocation can be classified into centralized and decentralized. For centralized revocation, a central entity, such as the CA, is the only entity in the network that can take the revocation decision for a certain node. For decentralized revocation, the node revocation is done by the neighboring nodes of the misbehaving node.

The rest of the paper is organized as follows: Section II presents the related work. The scheme description of our proposed scheme is introduced in Section III. Section IV introduces the performance metrics and the simulation environment. The simulation results are presented in Section V. Finally, Section IV concludes the paper.

II. Related work

In this section a review of revocation schemes for MANETs will be presented. In [4], the authors proposed an efficient decentralized revocation (EDR) protocol for vehicular ad hoc networks (VANETs). It is based on a novel pairing-based threshold scheme and a probabilistic key distribution technique which allow neighboring nodes to revoke malicious nodes. However, this scheme suffers from the heavily computations needed for the revocation process. In [5], the authors presented a decentralized certificate revocation scheme which utilizes certificates that are based on the hierarchical trust model. Their scheme delegates all key management tasks except the issuing of certificates to the nodes in a MANET; and it does not require any access to online certificate authorities (CAs). Their certificate revocation scheme is based on weighted accusations; whereby a quantitative value is assigned to an accusation to determine its weight. The weights of the accusations from nodes that are considered to be trustworthy are higher than those from less trustworthy nodes. A certificate of a node is revoked when the sum of the weighted accusations against the node is equal to or greater than a configurable threshold (RT).

Identity-based cryptographic (IBC) schemes have been considered to secure mobile ad hoc networks (MANETs) due to their efficient key management properties. However, these schemes do not provide mechanisms for key revocation and key renewal. In [6], authors propose the first key revocation and key renewal mechanisms for IBC schemes that are especially designed for MANETs. In their fully self-organized revocation scheme, each node monitors nodes in its communication range and securely propagates its observations. The public key of a node is revoked if a minimum number of nodes accused the node.

In [7], authors focused on the certificate revocation methods used in the certification system for MANETs. To cope with the wrong revocation of the certificate of legitimate users caused by false accusations by malicious nodes, and accordingly constructs clusters to detect false accusations. In [8], authors propose a distributed trust model for certificate revocation in Ad hoc networks. This model allows trust to be built over time as the number of interactions between nodes increase. Furthermore, trust in a node is defined not only in terms of its potential for maliciousness, but also in terms of the quality of the service it provides. Trust in nodes where there is little or no history of interactions is determined by recommendations from other nodes. Using elliptic curves for cryptographic protocols has been proposed in [9, 10]. Cryptosystems based on ECDLP can use smaller key size than that is needed by DLP or IFP based cryptosystems to provide the same level of secrecy. Reducing the key size while maintaining the same security level saves memory, computation power, and communication overheads which are major concerns in the resource constrains environment such as smart cards and MANETs.

III. SCHEME DESCRIPTION

In this section, we present an overview of the trust models and the system description of our proposed scheme.

A. Trust Models

There are three different models for building the trust in the mobile ad hoc network environment as follows:

1. Centralized Trust Model:

In this model, there is a well trusted entity known as a TTP. A TTP is an entity trusted by all users in the system, and it is often used to provide key management services.

2. Web-of-Trust Model:

In the web-of-trust model, there is no TTP that is well-trusted by all network nodes. Instead, peer nodes can issue certificates to each other and populate the certificate graph. Certificates can be authenticated through certificate chaining.

3. Decentralized Trust Model:

In MANETs, key management built on a fully centralized mode is not feasible because of the difficulty of maintaining and the central entity could become a single point of attacks. In the decentralized public key management scheme, the system public key is distributed to the entire network, while the system private key is split to multiple pieces (according to a secret sharing algorithm) and distributed to a subset (or all) of the nodes. The subset of group nodes creates a view of a CA and functions as a CA in combination. We use this model in our proposed scheme.

B. System description

The proposed protocol is based on both threshold cryptography and elliptic curve cryptography [11].

System Initialization:

Following are the notations of the parameters used in the system:

Fq: a finite field, $q>\max(n, SK)$, is a prime number.

E (Fq): an elliptic curve on finite field *Fq*.

G: a base point on elliptic curve *E* (*Fq*).

Ord (*G*) = : the order of the base point *G*.

SK∈*Zq:* the secret to be shared, it is a private key of CA.

: A secret share.

F(*x*): a polynomial.

: The coefficient of variable in the polynomial.

: A point on the elliptic curve.

Pk=*SK*⊙*G*: a public key related to *SK*.

The system is initialized as follows:

1- *CA* picks a secret polynomial ( ) $\sum$ ( ) and set *f* (*0*) =*SK*, it is the secret to be shared (private key of *CA*).

2- *CA* computes a secret share =*f* (u_i) (where u_i is the node identity and *i*=1,2,…, *n*) and sends it to each node as shown in Figure 1 through a perfect private channel which is safe enough to protect .

3- *CA* computes verify data = *G* (*j*=0, 1,…, *t*-1) and broadcasts it to the whole group.

4-When a node receives he checks if the equation $\sum$ is correct. If the test fails is an illegal data and is rejected.

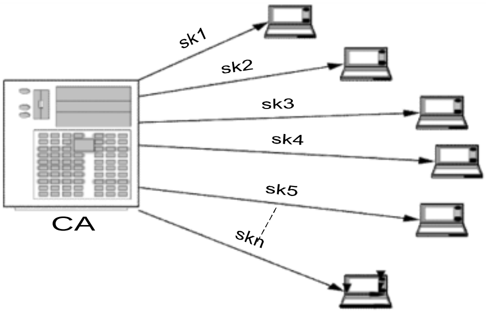


Fig. 1: CA distributes a secret share .

Revocation Process:

A misbehaving node can be revoked as follows.

- When a node exhibits misbehavior, one of the neighbors of the misbehaving node volunteers to take the role of the revocation coordinator.
- The revocation coordinator broadcasts to its one hop neighboring nodes a revocation request (RevReq) as shown in Figure 2 to share in the revocation process. It also sends a message *msg* containing the certificate of the misbehaving node, the reason for revocation, the current time stamp, the revocation coordinator signature on the entire message msg, and the revocation coordinator certificate.
- Any node receiving the RevReq and the message *msg* verifies the signature of the revocation coordinator on *msg* using the revocation coordinator's public key contained in its certificate and checks the time stamp to ensure the freshness of the message *msg*. If the verification succeeds the received node will send a revocation reply (RevRep) containing (i||) to the revocation coordinator as shown in Figure 2.
- When the number of RevREP's received by the revocation coordinator exceeds the threshold (*t*), the coordinator can reconstruct *SK* by using Lagrange Polynomial Interpolation as follows:

$$(\;)\quad \Sigma(\quad)\Pi\text{——}$$

If *x=0*, then the secret *SK* can be recovered by the formula

$$\Sigma(\quad)\Pi\text{——}$$

- At this point, the revocation coordinator is able to revoke any node by using *SK* which is identical to private key of *CA*

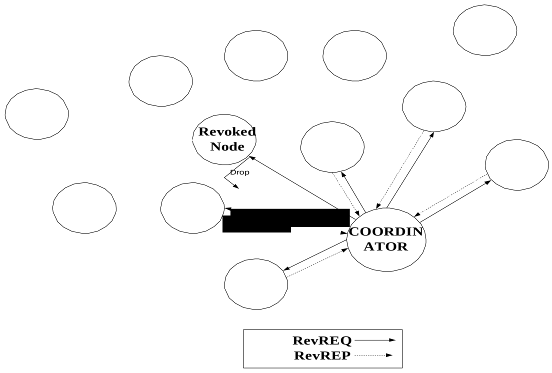


Fig. 2: Revocation request and reply

- The revocation coordinator broadcasts a certificate revocation result as shown in Figure 3 (RevRes) message = (msg||*T*stamp||sgncoord) to the neighboring nodes, where *T*stamp is the current time stamp, and sgncoord is the signature of the revocation coordinator on (msg||*T*stamp). Note that the certificate of the revocation coordinator is included in the message *msg*.

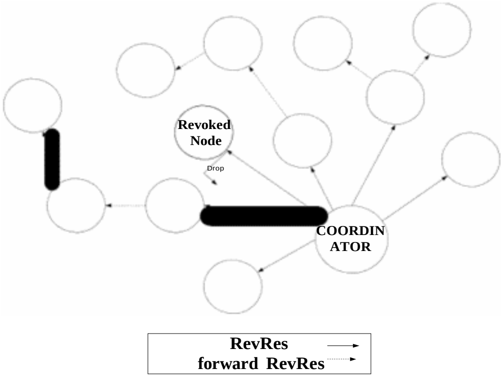


Fig. 3: Revocation result

- Any node receiving RevRes checks the freshness of the time stamp *T*stamp compared with that in *msg* to ensure that the revocation process is done in a timely manner, verifies the signature of the coordinator (sgncoord) using the coordinator's public key included in its certificate. If the verification of RevRes succeeds, it forwards the RevRes to other nodes in the network. The dissemination of RevRes continues until the lifetime of the revoked certificate ends. The Revocation process is summarized in the flow chart shown in Figure 4.

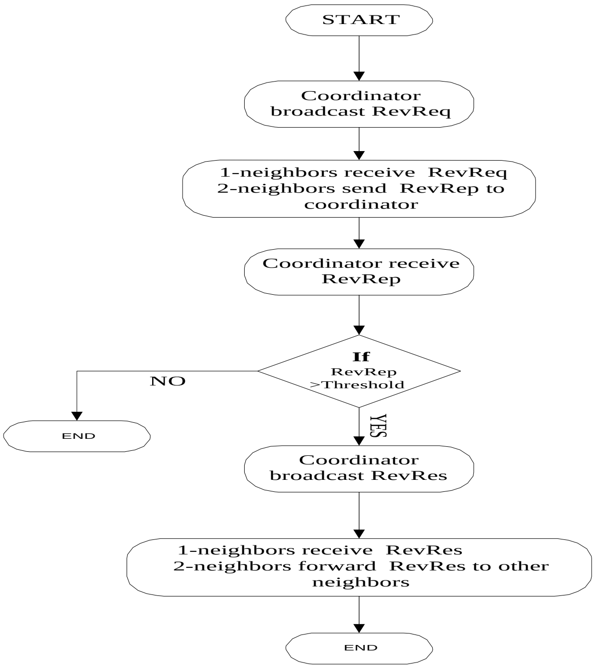

START
Coordinator broadcast RevReq
1-neighbors receive RevReq
2-neighbors send RevRep to coordinator
Coordinator receive RevRep
If RevRep >Threshold
NO
END
YES
Coordinator broadcast RevRes
1-neighbors receive RevRes
2-neighbors forward RevRes to other neighbors
END

Fig. 4: Revocation Process

IV.SIMULATION MODEL AND PERFORMANCE METRICS

A. SIMULATION ENVIRONMENT

Simulations were performed using Network Simulator (NS-2) [12], particularly popular in the ad hoc networking community. The MAC layer protocol IEEE 802.11 is used in all simulations. Nodes are spread randomly over the network. The NS-2 constant bit-rate (CBR) traffic generator is used to set up the connection patterns. Source initiation time is uniformly distributed over the first 10 seconds of the simulation time. Every simulation run is 500 seconds long. The *mobgenss* [13] mobility scenario generator was used to produce random mobility patterns. The pause time is set to zero. The Ad Hoc On-demand Distance Vector (AODV) routing protocol [14] was chosen for the simulations. The simulation results are the average of 10 runs. The threshold number of nodes (*t*) required to retrieve the *CA*'s private key (*SK*) varies from *5* to *20* in the simulation. The number of revoked nodes (*RN*) varies from *1* to *10* in the simulation. The rest of the simulation parameters are summarized in Table I.

B. PERFORMANCE METRICS

We have selected the Average Revocation Completion Time, Average Revocation Success percentage, and Certificate Revocation Percentage (CRL) as metrics during the simulation in order to evaluate the performance of the proposed scheme.
Average Revocation Completion Time: this is the average time required to complete the certificate revocation process.
Revocation Success percentage (*RSP*): The percentage of revocation reply (*RevRep*) delivered to the revocation coordinator for different values of the threshold (*t*).
Certificate Revocation List Update Percentage (*CRL*): The percentage of the number of nodes that received revocation results (RevRes) to the total number of nodes (*n*) in the network.

V. RESULTS

In this section, we present the main simulation results. We measure the Average Revocation Completion Time, Revocation Success percentage, and List Update Percentage (*CRL*) under different *RN* values. Figures 5, 6, 7 show the Average Revocation Completion Time versus threshold (*t*) for node mobility 0.1 m/sec, 5 m/sec, and 20 m/sec respectively. In order for the revocation process to be completed successfully, the revocation coordinator needs to receive at least *t* replies from his neighbors. The average revocation completion time increases with increasing the threshold (*t*) because with increasing *t*, the revocation coordinator needs to wait more time until he receives the required number of replies from his neighbors. For *t* equal to *20*, the coordinator cannot collect *20* replies from his neighbors and hence the revocation process fails and the average revocation completion time equal to zero. For *t* equal to 5 and 10, the node mobility has a slight impact on the average revocation completion time as the coordinator can easily collect the required *t* of replies.

Table 1: Simulation Parameters

Parameter	Value
Total number of nodes	50
Maximum number of connection	20
Area (m^2)	1000x1000
Radio transmission range	250m
Mobility Model	Random waypoint
Propagation Model	TwoRayGround
Mean speeds (m/s)	0, 5, 20
Data Rate	11 Mbps
Data packet size	512 bytes

The average revocation completion time is equal to zero when *t* is set to 15 for node mobility equal to 20 m/sec because it is very difficult for the revocation coordinator to collect 15 replies while the whole network is moving with this high mobility. Increasing the number of revoked nodes *RN* to 10 has a significant impact on completing the revocation process when the nodes move with mobility equal to 5 or 20 m/sec as shown in Figure 7. The revocation success percentage decreases with increasing the threshold *t* and it decreases with increasing the node mobility as shown in Figures 8, 9, 10. Increasing the number of revoked nodes in the network (*RN*) has a slight impact on the revocation success percentage as shown in Figures 8, 9, 10. Figures 11, 12, 13 show the Certificate Revocation List Update Percentage (*CRL*) versus threshold *t* for node mobility 0.1 m/sec, 5 m/sec, and 20 m/sec at *RN*=1, 5, 10 respectively. It is clear in Figures 11, 12, 13 that with increasing the threshold *t*, the *CRL* percentage decreases due to the difficulty of completing the revocation process when increasing the threshold *t* as explained previously in Figures 8, 9, 10. Increasing the number of revoked nodes in the network has a slight impact on the CRL percentage as shown in Figures 11, 12, 13.

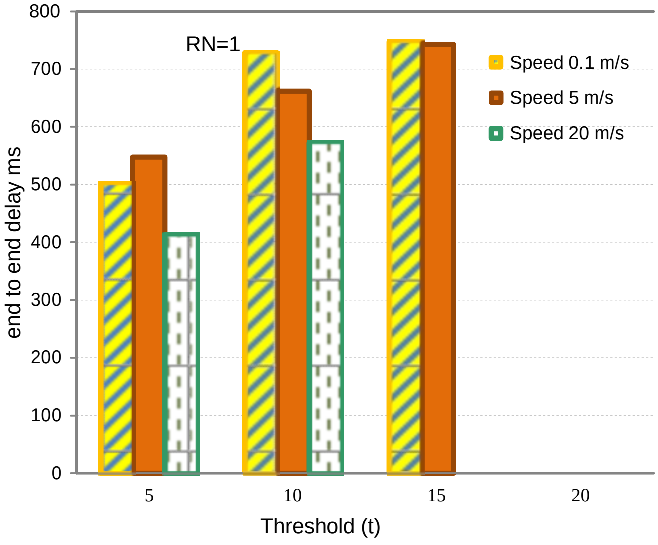


Fig. 5: end to end delay at RN=1

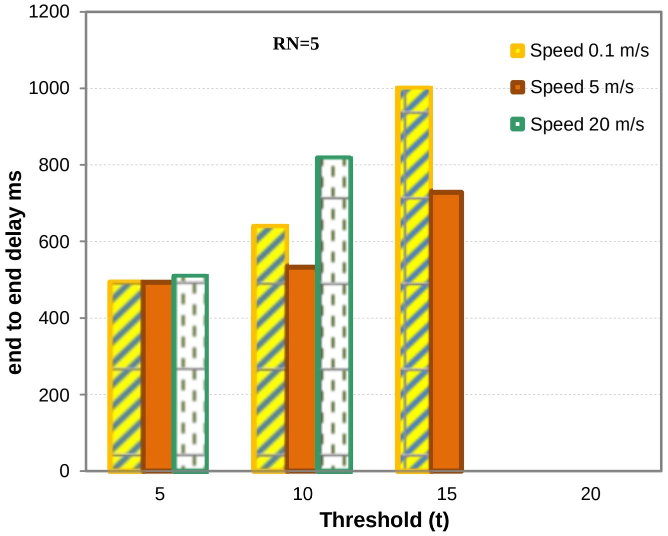


Fig. 6: end to end delay at RN=5

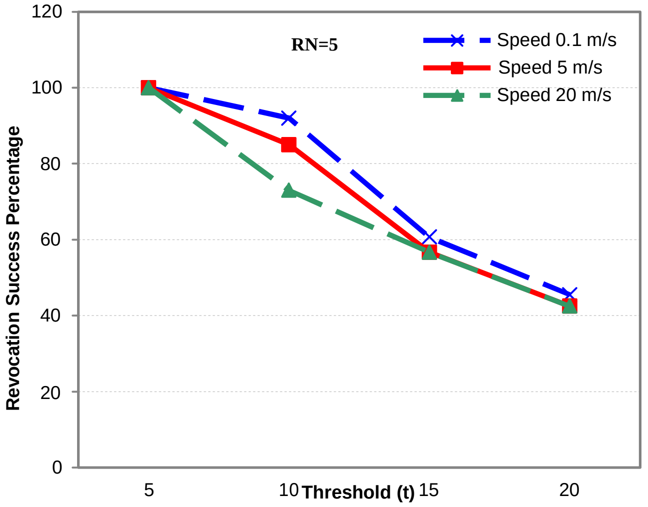


Fig. 9: Revocation success Percentage at RN=5

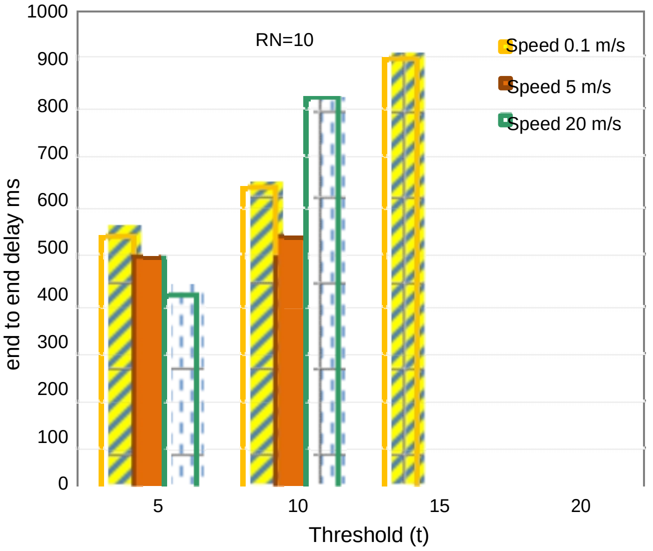


Fig. 7: end to end delay at RN=10

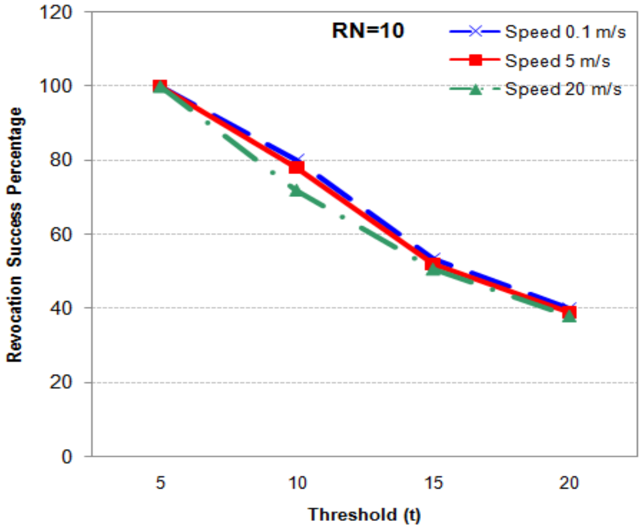
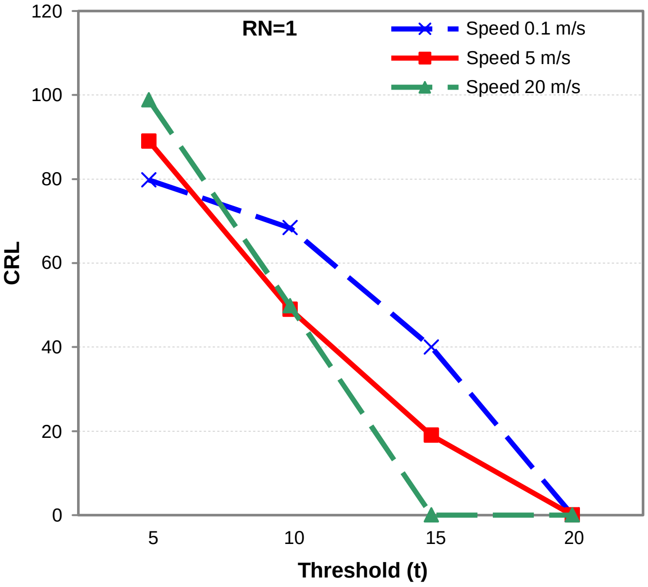


Fig. 10: Revocation success Percentage at RN=10



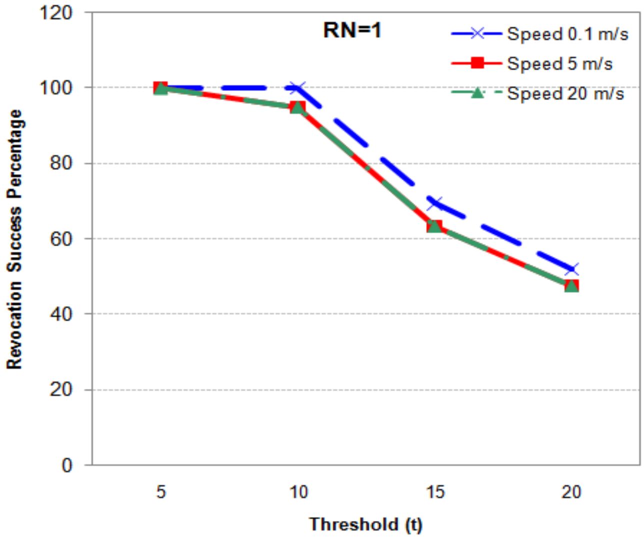


Fig. 8: Revocation success Percentage at RN=1

Fig. 11: Certificate Revocation Ratio at RN=1

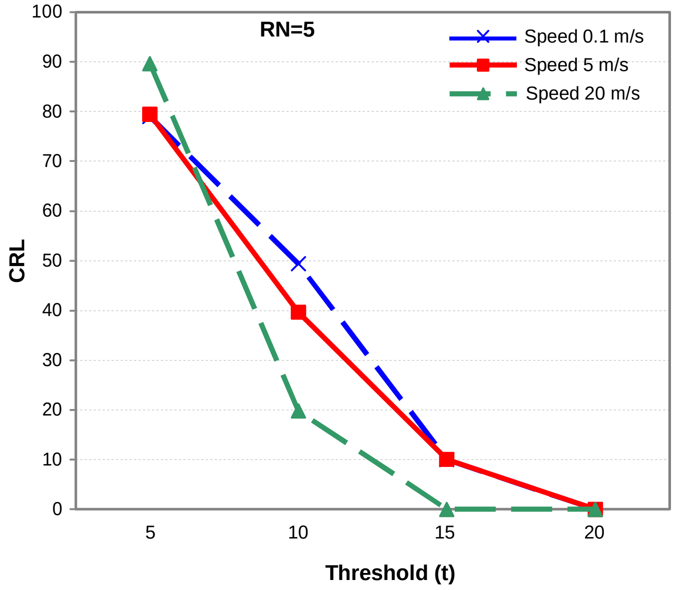


Fig. 12: Certificate Revocation Ratio at RN=5

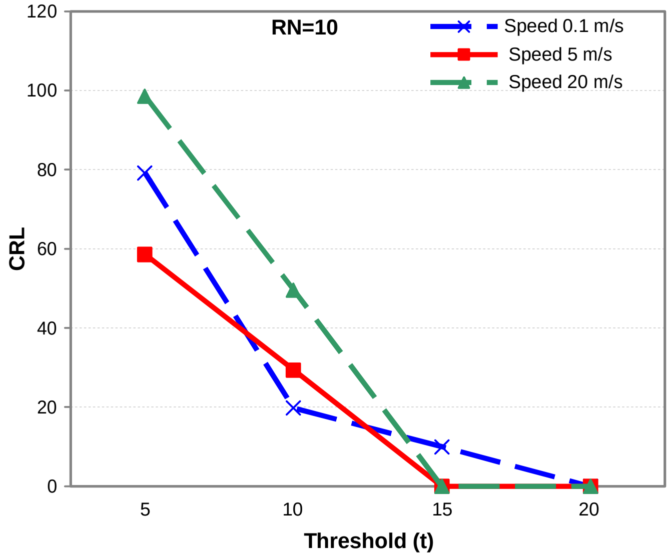


Fig. 13: Certificate Revocation Ratio at RN=10

VI. CONCLUSION

In this paper, we have proposed a revocation scheme for mobile ad hoc network (MANET) based on threshold cryptography. Our proposed scheme enables a group of legitimate nodes to perform fast revocation of a nearby misbehaving node. The simulation results show that when the threshold *t* is of reasonable value (10% to 20% of the total number of nodes in the network *n*), the proposed scheme can perform the revocation process successfully with high probability. Results also show that our proposed scheme does not suffer from increasing the number of misbehaving nodes in the network RN when performing the revocation process. The results show also that the proposed scheme is suitable for stationary and high mobility networks.

REFERENCES

[1] H. Dahshan, and J. Irvine, "Authenticated Symmetric Key Distribution For Mobile Ad Hoc Networks ", IEEE MASS 2008, Atlanta, Georgia, USA , September, 2008.

[2] H. Dahshan, and J. Irvine, " Key Management in Web of Trust for Mobile Ad Hoc Networks –, IEEE AINA 2009, Bradford, UK, May 26-29, 2009.

[3] H. Dahshan, and J. Irvine, " On Demand Self-Organized Public Key Management for Mobile Ad Hoc Networks –, IEEE VTC 2009 Spring, Barcelona, Spain, April 26–29, 2009.

[4] Albert Wasef, –Efficient Decentralized Revocation Protocol‖, , IEEE 2009. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 9, NOVEMBER 2009.

[5] Genevie`ve Arboit, and Claude Cre peau, –A localized certificate revocation scheme for Ad Hoc Networks‖, Elsevier Journal of Ad Hoc Networks, 2008, vol. 6, no. 1, pp. 17-31.

[6] Katrin Hoeper, and Guang 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, pp. 224-237, 2006.

[7] K. Park, H. Nishiyama, N. Ansari, and N. Kato, "Certificate revocation to cope with false accusations in mobile ad hoc networks," Proc. 2010 IEEE 71st Vehicular Technology Conference: VTC2010-Spring, Taipei, Taiwan, May 16-19, 2010.

[8] S. Chinni, J P Thomas, G Ghinea, Z Shen, –Trust Model for Certificate Revocation in Ad hoc Networks‖, Elsevier Journal of Ad Hoc Networks, 2008, vol. 6, no. 3, pp. 441-457.

[9] V. S. Miller, –Use of elliptic curves in cryptography,‖ Lecture notes in computer sciences; Advances in cryptology, pp. 417–426, 1986.

[10] N. Koblitz, –Elliptic curve cryptosystems,‖ Mathematics of Computation, vol. 48, pp. 203–209, Jan 1987.

[11] HAN Yiliang, –Verifiable Threshold Cryptosystems Based on Elliptic Curve‖, International Conference on Computer Networks and Mobile Computing (ICCNMC'03) , IEEE 2003.

[12] K. Fall and K. Vardhan, –The network simulator (ns-2).‖ Available at: http://www.isi.edu/nsnam/ns..

[13] W. Navidi and T. Camp, –Stationary distributions for the randomwaypoint mobility model,‖ IEEE Transactions on Mobile Computing, vol. 3, no. 1, pp. 99–108, IEEE 2004.

[14] C. E. Perkins and E. M. Royer, –Ad-hoc on-demand distance vector routing,‖ in 2nd IEEE Workshop on Selected Areas in Communication, (New Orleans, LA), 1999.