Improvement of Self-Organized Public Key Management for MANET

Marjan Kuchaki Rafsanjani [1], Bahador Shojaiemehr [2]

[1.] Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran
[2.] Science and Research Branch, Islamic Azad University, Kerman, Iran
kuchaki@mail.uk.ac.ir

Abstract: This paper studies key management, a fundamental problem in securing Mobile Ad hoc Networks (MANETs) and presents a description of a self-organized public key management scheme and a comparison among number of certificate-based authentication schemes for mobile ad hoc networks. In this paper we improve self-organized public key management by combine features of self-organized schemes.
[Marjan Kuchaki Rafsanjani , Bahador Shojaiemehr. Improvement of Self-Organized Public Key Management for MANET. Journal of American Science 2012;8(1):197-202-]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Authentication; Cryptography; Mobile Ad hoc Networks (MANET); Certificate chain; Public key management

1. Introduction

Mobile Ad hoc Networks (MANETs) are infrastructureless, Autonomous, stand-alone wireless networks that are receiving growing attention from both academia and industry. Authentication is one of the most important and challenging issues in MANETs. The scarcity of computation and communication resources and the lack of secure network infrastructures present major challenges for deploying applications in such a network.

The main problem of any public-key based security system is to make each user's public key available to others in such a way that its authenticity is verifiable. In mobile ad hoc networks, this problem becomes even more difficult to solve because of the absence of centralized services and possible network partitions. More precisely, two users willing to authenticate each other are likely to have access only to a subset of nodes of the network (possibly those in their geographic neighborhood). The best known approach to the public-key management problem is based on public-key certificates [1].

In MANET, key management can be classified into two kinds; the first one is based on a centralized or distributed trusted third party (TTP). The TTP is responsible for issuing, revoking, renewing, and providing keying material to nodes participating in the network such as situations where the key management process is performed using threshold cryptography. In the (n, t) threshold cryptography ,a secret key is divided into n shares according to a random polynomial and kept by n legitimate nodes, which we call share holders. Later, a new node needs to collect t shares from the response of t nodes (among n nodes) based on Lagrange interpolation and generates the original secret key as a legitimate node.

The second kind of key management is the self-organized key management schemes. 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 [2]. Certificates are stored and distributed by the nodes. Each node maintains a local certificate repository that contains a limited number of certificates selected by the node according to an appropriate algorithm. Public-key authentication is performed via chains of certificates.

In this paper, we improve self-organized public key management by combine features of self-organized schemes and compare them. The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 provides requirements of effective certificate-based authentication for MANETs. In section 4, we describe essential quantity to generate trust relationship between nodes. The restrictions and problems of first scheme by Capkun and self-organized assumptions are described in sections 5 and 6. The system description and trust model of our proposed scheme are presented in section 7. Scheme efficiency parameters and comparison of the schemes is described in Section 8. Finally, Section 9 concludes the paper.

2. Related Work

In this section, a review of key management schemes for MANETs will be presented. Capkun et al. [1] proposed a self-organized public key management scheme in which each node issues certificates independently and manages them at its repository. In this scheme, certificates are stored and distributed by the nodes and each node maintains a local certificate repository that contains a limited number of certificates selected by the node according to an appropriate algorithm. Key authentication is

performed via chains of certificates. However, this scheme suffers from the delay and the large amount of traffic required collecting the certificates.

In [2], the proposed scheme relies on establishing a small number of trust relations between neighboring nodes during the network initialization phase. Experiences gained as a result of successful communications and node mobility through the network enhance the formation of a web of trust between mobile nodes. The proposed scheme allows each user to create its public key and the corresponding private key, to issue certificates to neighboring nodes, and to perform public key authentication through at least two independent certificate chains without relying on any centralized authority. A measure of the communications cost of the key distribution process has been proposed.

In [3] has presented a description and performance evaluation of a threshold secret sharing (TSS) authentication scheme for self-securing mobile ad hoc networks (MANETs) suffering from high packet-loss and node mobility. Authors in order to evaluate the performance of their TSS scheme in a noisy MANET, a number of simulations were carried-out. They concluded that presence of noise inflicts significant reduction in the authentication success ratio (S_R) and consequently degrades the performance of the network, while node mobility inflicts no or insignificant effects.

Wang et al. [3] uses threshold cryptography for public key management. The system CA (Certificate Authority) key pair is denoted as {*SKR, SKU*}, where *SKR* is the system private key and *SKU* is the system public key. *SKR* is used to sign certificates for all nodes in the network. A certificate signed by *SKR* can be decrypted only by the well-known public key *SKU*. In a TSS scheme, *SKR* is shared among network nodes. Each node n_i holds a secret share SKR_i, and any k of such secret share holders can collectively function as the role of CA. However, for better system security, the secrecy of *SKR* is preserved all the time and it is not visible, known or recoverable by any network node.

3. Requirements of Effective Certificate-Based Authentication for Ad Hoc Networks

Five requirements have been identified for any certificate-based authentication scheme to be considered *secure* and *effective*, with respect to authentication in a mobile ad hoc network.

Requirement 1) *Distributed authentication*: In ad hoc networks, due to issues such as frequent link failures, node mobility, and limited wireless medium, it is typically not feasible to include a fixed centralized CA in the network. Further in networks requiring high security, such a server could become a single point of failure. For example, consider a battle field scenario, where the troops are spread over a large area. In such a case, it might not be feasible to have a central server. Consider an enemy attack on the server - this would bring down the whole network! One of the primary requirements of a certificate-based mechanism is to distribute the authentication amongst a set of nodes in the network.

Requirement 2) *Resource awareness*: Since the nodes in an ad hoc network typically run on batteries with high power consumption and low memory capacity, the authentication protocols must be resource-aware. That means the time and space complexity of the underlying algorithms must be acceptably low. In this regard, symmetric-key-based cryptographic techniques are more suited, as compared to public key methods, since symmetric cryptography in general incur less resource consumption. However, the issue of distributing the symmetric keys prevents their practical deployment in ad hoc networks. This is a tradeoff that must be dealt with at the application level. Since the certificate-based authentication uses public key mechanisms, which are resource-intensive, the protocol itself must be efficient both in terms of memory and power.

Requirement 3) *Efficient certificate management mechanism:* The distribution of public keys and management of certificates have been studied extensively in the case of wired networks. However, in applying these methods to MANETs, managing the certificates (creation, revocation and renewal) is a challenging issue. We discuss this further in Sections 3 and 4. Most of the current mechanisms lack a robust certificate revocation scheme.

Requirement 4) *Heterogeneous certification:* As in the case of wired networks, the certifying authorities might be heterogeneous even in ad hoc networks. This means that two or more nodes belonging to different "domains" may try to authenticate each other. In such a case, there must be some kind of trust relationship or hierarchy among the Certifying Authorities. In wired networks, this is accomplished through certificate chaining.

Requirement 5) *Robust pre-authentication mechanism:* By pre-authentication mechanism we mean the process of establishing necessary trust between nodes before the actual certificate creation and distribution. Though this is not a part of the certificate authentication process itself, it is pretty important in MANETs. This is because, in order to satisfy R.1, it is mandatory that nodes have prior trust between each other (by exchange of public keys, for example). Without this established, the later mutual

authentication and renewal of certificates would not be possible.

4. Trust Evaluation

Each node that wishes to join the network can establish independent trust relationship with some of the existing member nodes in the network. For example, a node that wishes to join the network contacts one of the existing network members through secure side channels and provides its trust evidence. If the existing network member believes that the requesting node is trustworthy according to its trust evidence, they can sign and exchange certificates. The process is repeated until the joining node gets a sufficient number of certificates [2].

Trust value represents the assurance with which a requesting node can obtain the correct public key of a target node. However, the same assurance in the reverse direction needs not to exist at the same time. In other words, the trust relationship is unidirectional. Each node in the network should have a trust table as shown in Table 1 to store the public-key certificates and the corresponding trust values of the nodes it trusts in the network. There are many trust metrics have been proposed to evaluate the trust values, some assume discrete trust values as in PGP. Others assume continuous values for trust [3]. In our trust model, we define the trust value as a continuous value between 0 and 1.

During the network initialization phase, neighboring nodes exchange the trust evidence (for example, driver license, passport, employment identity card, date of birth, and documentation indicating credit card activity) and according to the validity and the strength of the exchanged evidence, mobile nodes are able to assign trust values for each others before certificates exchange process. A trust value $T_{i,j}$ represents node i's belief that node j is trustworthy. The higher the value of $T_{i,j}$, the more node i trusts node j, and *vice versa*. Any node in the network can calculate the value of trust $T_{i,j}$ in another node's public key if there exist a certificate chain between the two nodes using formula 1 [2].

$$T_{i,j}=\prod_{K=1}^{K=h} T_k \qquad (1)$$

where T_k is the value of trust between two directly trusted nodes along the certificate chain from node i to node j, and h is the number of hops between node i and node j.

5. Restrictions and Problems of Capkun's Scheme

The authentication scheme is just suitable for small or medium scale MANETs which consist of tens or at most hundreds of mobile nodes in a small area. The distributed schemes can't ensure high service availability, and the chained authentication's randomicity makes it not secure enough.

Therefore in the first scheme by Capkun et al. if node density is more than specified measures, the nodes unable to maintain certificates in local repositories. Moreover when exchange certificates with neighbor nodes to prevent malicious nodes must be checks consistent of the certificates and determines which user-key bindings are correct. It means that all of the issued certificates for specified node should be contain the same usernames and public keys. This mechanism needs time consuming process and causes network delay.

The later schemes self-organized public key management proposed improves these defects in the first scheme.

6. Self-Organized Schemes Assumptions

One of the assumptions in the self-organized public key management mechanisms are that network is not scalable. These schemes used to small or medium scale MANETs.

In [2], if the numbers of hops between source and destination nodes are many, then because insertion of the certificates per hop, the route request packet is large and causes traffic of the network and delay of the authentication.

Another assumption is that mobile nodes are most likely to be stationary or moving with low mobility in order to exchange the trust evidence and hence establish trust relations.

In [2], problems are delay and traffic because:

- Establish of certificate chain is performed when authentication system is active. In other words routing data packet and authentication are not disjoint.
- Intermediate node multi-cast packet contains certificates to all the neighbor nodes. Some of the nodes not provide independent route to destination. Therefore destination ignores such that nodes.

To improve second problem use below function:

In the given network in figure 1, three routes exist between node 1 and 6:

Route1:1, 2, 6.

Route2:1, 5, 6.

Route3:1, 2, 3, 4, 6.

Two certificate chain 1, 2 identified by destination and they are independent because do not have any common intermediate nodes. Two

certificates chain 1, 3 is dependent and establish certificate chain 3 just cause traffic in the network.
Each node in the network should have a trust table as shown in Table 1 to store the public-key certificates and the corresponding trust values of the nodes it trusts in the network. There are many trust metrics have been proposed to evaluate the trust values, some assume discrete trust values as in PGP. Others assume continuous values for trust. In trust model the trust value is a continuous value between 0 and 1.

Table 1. Trust table

Node ID	Certificate	Trust value

During the network initialization phase, neighboring nodes exchange the trust evidence (for example, driver license, passport, employment identity card, date of birth, and documentation indicating credit card activity) and according to the validity and the strength of the exchanged evidence, mobile nodes are able to assign trust values for each others before certificates exchange process.

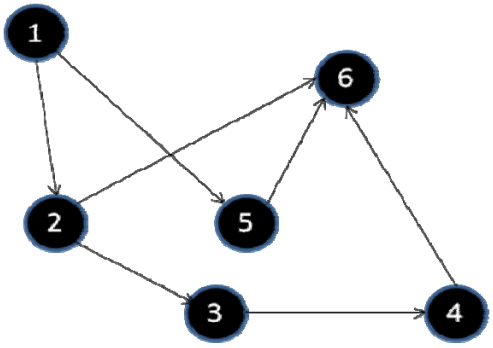

Figure 1. Instance of graph certificate

Solution is that each node to forward route request packet first in to route request table checks identity of route request packet. If packet with same identity forward before, node does not sends new route request packet and discard it.

In the mentioned network, node 2 when sends route request packet to node 6, it does not send packet to node 3.

Apply this functionality in scheme can decrease traffic of the network considerably.

In our proposed scheme we try to remove or minimize time of certificate chain establishment by using certificate exchanges in initial phase.

7. Our Scheme Description

- Each node creates its own public/private key pair.
- Each node use side channel (over an infrared channel at a time of a physical encounter) to provide its trust evidence for neighbor nodes. If neighbor nodes believe trustworthy, then establish trust relationship. Then via this channel trust nodes exchange public keys.
- Each node in its local repository only the certificates issued and the certificates that other nodes issued to it. In this way, each certificate is stored at least twice: by its issuer and by the user to whom it is issued.
- Nodes based on memory space to maintain know how many certificates can be maintained. Node multi-cast set of its certificates to neighbor nodes. The process of sending the certificate will continue since the entire network reach of all the certificates or assigned memory space of the nodes gets full.
- Each node searches its memory compares ID of certificates that maintained in memory with ID of received certificate. If certificate is repetitive then certificate is marked and node prevents to send it for neighbor nodes.
- When the nodes constructed its certificate repositories, they are ready to perform authentication. If the repository of the node contains entire network certificates then the node able to authenticity of destination public key by tracking certificate chain locally, else the node contains part of certificates in the entire network. Such that node create certificate route to destination by merge local certificate repository with certificate repository of intermediate nodes.

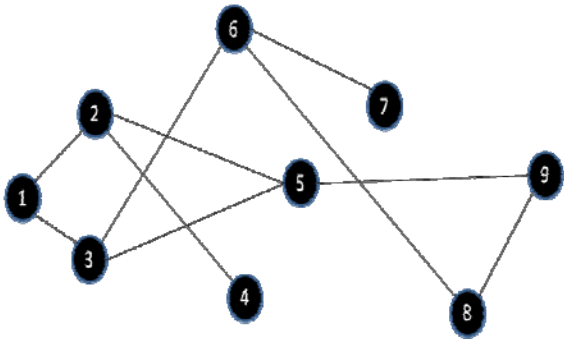

Figure 2. Repository of nodes before the certificate exchange

In the example shown in figure 2, the maximum of certificates that nodes can maintain are six. Table 2 shows maintained certificates before the certificate exchange process.

In the Table 2, *(i,j)* refers to certificate that node *i* issued to node *j*.

Table 2. Certificate repositories before certificate exchange

Node no.	Certificates					
1	(1,2)	(1,3)	(2,1)	(3,1)	-	-
2	(2,1)	(2,4)	(2,5)	(1,2)	(4,2)	(5,2)
3	(3,1)	(3,6)	(3,5)	(1,3)	(6,3)	(5,3)
4	(4,2)	(2,4)	-	-	-	-
5	(5,2)	(5,3)	(5,9)	(2,5)	(3,5)	(9,5)
6	(6,3)	(6,7)	(6,8)	(3,6)	(7,6)	(8,6)
7	(7,6)	(6,7)	-	-	-	-
8	(8,6)	(8,9)	(6,8)	(9,8)	-	-
9	(9,5)	(9,8)	(5,9)	(8,9)	-	-

After the certificate exchange, if node 1 gets two certificate (2,5) and (2,4) from node 2 and want to authenticate node 9, this action performed by merge its repository and repository of node 5 (see figure 3).

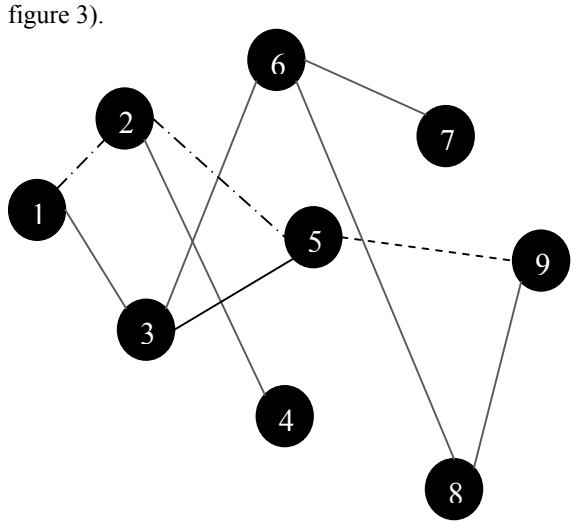

Figure 3. Certificate route between nodes 1and 9

8. Trust and Clustering Based Schemes for MANET

In this section, we describe trust and clustering-based authentication. The network model is based upon hierarchical organization or clustering of the network by some clustering algorithms. The authors of the method perceive that such algorithms improve the security and the efficiency of the network. They assume that the network has been divided into clusters with unique IDs.

The network model is based upon hierarchical organization or clustering of the network by some clustering algorithms. Authors of the method perceive that such algorithms improve the security and the efficiency of the network. They assumed that the network has been divided into clusters with unique IDs.

Their trust model is based upon the web-of-trust model similar to PGP, in which any user can act as the certifying authority. They define trust quantitatively as a continuous value between 0 and 1. Each node maintains a list of trust values for other nodes in the network. A *direct trust* is defined as a trust relationship between two nodes in the same group, and a *recommendation trust* as the trust relationship between nodes of different groups. In order to build the trust relationship they assume that the nodes are equipped with some detecting component such as watchdog for monitoring the behavior of nodes.

Public key management is assumed to be present within a cluster. Whenever a node wants to authenticate a node in another cluster, it communicates with several other *introducing nodes* in that cluster. It sorts the *introducing nodes* based on their trust values and computes a weighted trust value by combining its trust values of the *introducing nodes* with the trust values of the *introducing nodes* to the target node. The final trust value is then stored and used to evaluate other nodes in that group. The advantage of the mechanism is that it is able to discover and isolate a high percentage of malicious nodes when compared to PGP based methods. Disadvantage is that the storage of the trust values and their computation is both memory and time consuming. The mobility of nodes leads to change of membership of nodes in various clusters.

9. The Comparison of Public Key Management Schemes for MANET

In Table 3, the five mechanisms are compared with respect to the requirements described earlier.

10. Conclusions

This paper demonstrate that self-organized public key management schemes and combine them to improve efficiency and describes requirements of effective certificate-based authentication for ad hoc networks. Then it compares certificate – based authentication schemes by the mentioned requirements. In this improved scheme two users in a mobile ad hoc network can perform key authentication based only on their local information, even if security is performed in a self-organized way.

Table 3. Comparison of certificate-based authentication methods

Methods / Requirements	Self-organized Capkun	Self managed heterogeneous	Trust and clustering based	Threshold cryptography	Proposed scheme
Distributed authentication	Distributed completely since every node acts as a CA	Distributed completely and suitable for large scale networks	Distributed completely since every node acts as a CA	Distributed since all or group of the nodes act as CA	Distributed completely since every node acts as a CA
Resource awareness	Every nodes have high process and maintain two repository which incurs a high overhead	Every nodes only maintains a list of its trusted CA's. thus resource consumption is efficient	The maintain of trust tables and the monitoring components are memory intensive	Needs only process to generate CAs private key	The maintenance of trust tables and certificate repository
Certificate creation	Certificates issued by collaboration between nodes	Requires at least K neighbors to generate partial signatures	Certificate creation based on trust values	Requires at least K neighbors to generate partial signatures	Certificate creation based on trust values
Heterogeneous certificate	Not implemented	Implemented using trust graphs	Not implemented	Not implemented	Not implemented
Pre-authentication	Identify neighbor nodes and certificate repository creation	Identify neighbor nodes and create the list of trusted CAs	Establish trust relationships and certificate repository creation	Determine authentication servers in the network	Establish trust relationships and certificate repository creation

Corresponding Author:
Marjan Kuchaki Rafsanjani
Department of Computer Science
Shahid Bahonar University of Kerman
Kerman, Iran
E-mail: kuchaki@mail.uk.ac.ir

References

1. Capkun S, Buttyan L, Hubaux J-P. Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing 2003:52-64.
2. Dahshan H, Irvine J. A robust self-organized public key management for mobile ad hoc networks. Security and Communication Networks 2010;3:16-30.
3. Wang G, Wang Q, Cao J, Guo M. An effective trust establishment scheme for authentication in mobile ad hoc network. International Conference on Computer and Information Technology 2007:749-754.
4. Edith CH, Ngai S, Lyu MR. Trust and clustering-based authentication services in mobile ad hoc networks. Distributed Computing Systems Workshop 2004:582-587.
5. Castelluccia C, Yi JH. Robust self-keying mobile ad hoc networks. Computer Networks 2007;51:1169-1182.
6. Weimerskirch A, Thonet G. A distributed light-weight authentication model for ad hoc networks. 4th International Conference on Information Security and Cryptology 2001:341-354.

3/5/2011