

ANALYSIS PARAMETERS AND DISTRIBUTED AUTHENTICATION IN MOBILE AD HOC NETWORKS

M.VINOTH¹

Research scholar, ECE Department, SCSVMV
University
vinoth24@gmail.com.

S.OMKUMAR²

Associate Professor, ECE Department, SCSVMV
University
Omkumar1234@gmail.com.

ABSTRACT

The distributed authentication is well studied in wired networks. However, adapting certificate-based authentication protocols to mobile ad hoc networks (MANETs) is a nontrivial task, mainly because, in a MANET, as opposed to conventional wired networks, there typically exists no fixed infrastructure or centralized management. For example, a conventional certificate-based authentication system relies on a fixed trusted Certificate Authority (CA), which is responsible for the creation, distribution, renewing, and revocation of certificates. In a MANET, due to issues such as node mobility, limited wireless medium, and frequent link failures, it is typically not feasible to include such a fixed centralized CA in the network. Various approaches have been proposed to tackle the unique challenge of adapting certificate-based methods for distributed authentication in mobile ad hoc networks. Our contribution in this paper is twofold: we first analyze the requirements of a secure distributed authentication system for MANETs, and then survey some of the existing certificate-based authentication mechanisms, by analyzing their features, including pros and cons, in the context of distributed authentication. Finally, a series of scenario-based simulation experiments and metrics are proposed to evaluate these features.

KEY WORDS

Ad-hoc and sensor networks, authentication, evaluation, simulation

1. Introduction

Mobile Ad Hoc Networks (MANETs) have received drastically increasing interest, partly owing to the practical applicability of MANETs to myriad applications. The deployment of such networks, however, poses several challenging issues, due to the dynamic nature of the nodes, the arbitrary topology, the limited wireless range of nodes, and transmission errors. Since all the nodes in the network collaborate to forward the data, the wireless channel is prone to active and passive attacks by malicious nodes, such as Denial of Service (DoS),

eavesdropping, spoofing, etc. Implementing security is therefore of prime importance in such networks.

The five components of a security mechanism are confidentiality, integrity, authenticity, availability and non-repudiability. Out of these, authenticity is the most fundamental issue, since a breach of authenticity leads to a system-wide compromise. One of the widely used authentication mechanisms in conventional wired networks is the public key management system using certificates.

One of the main issues to consider in a certificate-based scheme is the secure distribution of the public keys to all the nodes in the network. The *Public Key Infrastructure (PKI)* [1] defines methods to handle public key management using X.509 certificates. In a wired network, there exists a centralized certificate server which handles the creation, renewal and revocation of certificates. This is not feasible in ad hoc networks, due to the absence of a fixed infrastructure and centralized management. Besides, due to the dynamic topology of the network, frequent link failures may occur, resulting in issues such as re-authentication and timely communication with the certificate server.

To overcome these limitations and to reap full advantages of the certificate-based authentication mechanism, several public key management mechanisms have been proposed [2] [3] [4] [5] [6] [7]. In this paper we analyze some of these methods, and discuss their pros and cons. The rest of the paper is organized as follows. Section 2 discusses the requirements of a certificate-based authentication scheme for mobile ad hoc networks. Section 3 provides a survey and brief description of the employed mechanisms. In Section 4 we compare the schemes with respect to the requirements. In Section 5 we enumerate scenarios and metrics for the simulation study of these mechanisms.

2. 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.

R.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.

R.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.

R.3 Efficient certificate management mechanism: The distribution of public keys and management of certificates have been studied extensively in the case of wired networks [1]. 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.

R.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.

R.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. The Resurrecting Duckling Model proposed by Stajano and Anderson [8] was one of the early works in this field, which involved bootstrapping trust between a “mother” and a “duckling” node over a location-limited channel. Balfanz et al [9] discuss a more user-friendly and efficient approach. A detailed classification of these methods is beyond the scope of this paper.

3. Survey of Related Work

Certificate-based authentication usually consists of three phases. During the first phase or the “bootstrapping” phase, the nodes are issued a certificate by a certifying authority. The certificate is created by the CA using the node’s identity information, such as IP address, name, organization, and its public key. The certificate also consists of the issuing time and the expiration time besides other information. During the second phase the certificate is “renewed” due to its expiration. The third phase involves revocation of the certificate by the CA, possibly due to compromise of the private key of the certificate holder, or probably because the issuer believes that the user-key binding is no longer valid. We now discuss some of the proposed mechanisms.

3.1. Self organized public key management

One of the certificate-based authentication methods proposed by Capkun, Buttyan and Hubaux is by formation of certificate graphs [2]. The suggested approach is similar to PGP certificates [10], apart from the fact that in PGP a central certificate server is used. A certificate graph is defined as a directed graph $G(V, E)$ where V and E stand for the set of vertices and the set of edges, respectively. The vertices of the certificate graph represent public keys, and the edges represent certificates. As shown in Figure 1, a directed edge in the graph from vertex K_u to K_v represents the certificate issued by u to v by u 's signing v 's public key K_v with its own private key. In effect, thus, u is the CA for v . G contains only the valid certificates of the whole network.

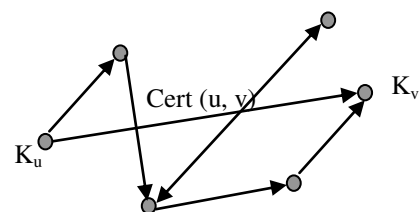


Figure 1: $K_u \rightarrow K_v$, certificate issued to v by u

Each node maintains an updated and non-updated local certificate repository, which consist of subset of updated and expired certificates respectively. Capkun et al argue that the use of two repositories is in providing a good estimate of the certificate graph and for node authentication. Whenever a user u wants to verify the authenticity of the public key of another user v , u tries to find a directed path in the graph by merging the updated certificate repository graphs of u and v . The chain of certificates on the path is used to authenticate v . If no path is found then the node merges its non-updated and updated certificate repositories to find expired certificates in the path. On finding such a path, it updates the expired certificate, checks the correctness and performs authentication.

The certificate creation phase begins by every node generating its own public-private key pairs. When a new node requests for a new certificate from its neighbor, the issuer verifies the authenticity of the public key. Capkun et al assume that this is done by pre-exchanging their keys over a side channel. In order to update the certificate graphs in the updated repository, a certificate exchange phase is carried out by exchanging hashes of the certificates with neighboring nodes periodically. There is an upper bound on the convergence times before all the nodes get updated with the certificate graphs. In order to maximize the efficiency of the *updated certificate repository* creation and updating, Capkun et al propose algorithms such as *Maximum degree algorithm* based on finding the path in the certificate graph with highest number of certificates.

Capkun et al do not mention any explicit certificate renewal process as it is done whenever a node finds expired certificates in its non-updated certificate repository. They suggest two methods, one explicit and the other implicit, for revocation of the certificates. In the *implicit* mechanism, the certificates are revoked based on their expiration time. In the *explicit* method, the issuer sends an explicit revocation statement for the target node that it believes no longer has a valid user-key binding. This is sent to nodes that request the issuer for updates of the certificate for the target node.

The advantage of this mechanism lies in the fully self-organized management of public keys by using certificates. However the drawbacks of this scheme are the expensive tables that have to be maintained for the certificate repositories, and each time a node moves from one locality to another, it has to renegotiate with other nodes and update the tables again.

3.2. Providing Robust and Ubiquitous Security Support for MANETs

In this scheme, Kong et al [4] propose a distributed certification based on threshold cryptography and shared secrets. The basic goal of a threshold secret sharing

method is to share a secret key k among an arbitrarily large community using a secret polynomial $f(x)$. If the degree of $f(x)$ is $(k-1)$, any k members of the community can recover the secret key, while any members less than k reveals no information of the secret [6]. Based on this, a node receives its public key from its k neighboring nodes. Here, k is a parameter which needs to be carefully tuned so that the method is effective.

The certificate creation process is as follows: Initially all the nodes in the network need to be bootstrapped with their certificates from a trusted central management. When a new node wants to obtain its certificate, it sends a request to its k neighboring nodes requesting for partial certificates. If the coalition thinks that the requesting node is a well-behaved node, they issue their partial certificates, which are then combined together by the target node to issue the new certificate using an interpolation function.

The certificate renewal is carried out by specifying a renewal Time T_{renew} . To renew a certificate, a network entity broadcasts its current valid certificate and a future expiration time $T < (\text{current time} + T_{\text{renew}})$ to its k one-hop neighbors. The neighboring nodes check the system public key and the Certificate Revocation List to determine whether to accept or deny the request.

The certificate revocation is carried out by two methods as suggested in [2] by implicit or explicit mechanisms. In the implicit mechanism, the certificates are revoked if the expiration time (T_{expire}) is lesser than the time of issue plus the time of renewal (T_{renew}). In the explicit certificate revocation method, each node maintains a Certificate Revocation List containing those certificates that haven't expired yet. The node periodically consults its CRL for expired certificates and revokes them if necessary.

The basic advantage of this method is that it does not require any centralized certificate authority. However, it relies on each node having at least k one-hop neighbors for authentication. This may not be practical when k is large due to the dynamic nature of the nodes. Further, the certificates cannot be issued to nodes which are more than a hop away. It also requires a bootstrapping phase in order to distribute the system private key among k nodes initially.

3.3. Self Managed Heterogeneous Certification

Wang, Zhu and Li [3] propose a novel mechanism in which CAs from different administrative domains can co-exist in the network. They also propose a distributed certificate authority by using k -threshold secret sharing similar to the method introduced by Kong et al [4]. In order to handle heterogeneous CAs, trust graphs are used. A node A is said to trust node B when node B can be verified as authentic based on B 's digital certificate signed

by a CA that *A* currently trusts. Each node maintains a list of CAs that it trusts.

Whenever a node needs to obtain a certificate, it has to collect *K* IDs of valid share holders from its one-hop neighbors and constructs the private key. Whenever a node *A* wishes to authenticate another node *B*, it begins by sending *B* its CA list. Similarly *B* sends *A* its own CA list. *A* then compares the two lists to check if there are some common CAs, and if so, *A* proceeds to send its certificate to *B* certified by the common CA. *B* responds by sending its own certificate to *A*. If the two nodes don't have a common CA, then they proceed to search their one-hop and two-hop neighbors through a *Distributed Multi-hop Certificate Request* (DMCR) algorithm.

The steps for certificate renewal are similar to the DMCR scheme. However certificate revocation is not discussed.

Main advantages of this approach include (i) support for cross-certification between CAs in different domains; (ii) the certificate discovery mechanism occurs over *multiple-hops*.

3.4. Trust- and Clustering-Based Authentication

Ngai et al [5] discuss a trust model and a network model in order to enhance the security of public key certification. Their network model is based upon hierarchical organization or clustering of the network by some clustering algorithms. The authors 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.

Their trust model is based upon the web-of-trust model similar to PGP [10], 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 authors do not discuss a mechanism for renewal and revocation of the certificates. 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. The disadvantage is that the storage of the trust values and their computation is both memory and time consuming. Further, the mobility of nodes leads to change of membership of nodes in various clusters.

4. Comparison of the Mechanisms

In Table 1, the four mechanisms are compared with respect to the requirements described earlier. We do not consider requirement R.5, since it is not an inherent part of the certificate mechanism itself.

Table 1: Comparison of Certificate-based Authentications

Requirements	<i>Self Organized Public Key Management - Capkun</i>	<i>Providing Robust and Ubiquitous Security Support for Mobile Ad hoc Networks – Kong</i>	<i>Self Managed Heterogeneous Certification in Mobile Ad Hoc Networks – Wang</i>	<i>Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks – Ngai</i>
R.1. Distributed authentication	It is a totally distributed certification method since every node acts as a CA.	Totally distributed and scales well to large networks	Totally distributed and scales well to large networks	Distributed and self organized since every node acts as a CA
R.2. Resource awareness	Each node maintains two certificate repositories, which incurs a high overhead.	The generation and distribution of keys using complex polynomial functions is resource-intensive and time consuming.	Each node only maintains a list of its trusted CAs. Thus it is more efficient than method proposed in [2].	The maintenance of trust tables and the monitoring components are memory intensive.

R.3.(a) Creation	Self-signed certificates, and hence more robust than a shared key based mechanism	Requires at least k neighbors which might be a bottleneck	Similar to K-threshold mechanism [4]	Across nodes, creation is based on trust values. The existence of introducing nodes may not be true at all times.
R.3.(b) Renewal	No explicit mechanism discussed	Same as issuance	Implemented through the DMCR algorithm	Not discussed
R.3.(c) Revocation	Explicit revocation causes delay between far-away nodes in the network.	System CRL table stored at each node and hence memory intensive.	Not discussed	Not discussed
R.4. Heterogeneous certification	Not implemented.	Not implemented.	Implemented using trust graphs.	Not implemented

5. Scenarios and Metrics

In order to study the effectiveness of these mechanisms, we propose a set of realistic “scenarios” for simulation. Before defining the scenarios, we first need to define some parameters.

5.1. Parameters for defining the scenarios

1) The *mobility model* represents the realistic movements of nodes in the network. They can be primarily classified as entity mobility models and group mobility models. Camp et al. give a broader classification of these models [11]. The most commonly used mobility model by the research community is the RWM (Random Waypoint Model) which uses pause times and random changes in destination and speed. However, the randomness doesn't suit well to certain scenarios such as a battlefield, where the mobility is more predictive. Further, the model also fails to provide a “steady-state” over a long simulation period [12]. Thus, the mobility models should be chosen carefully while evaluating a certificate-based authentication mechanism. It must model the realistic scenario as closely as possible.

2) *Node Density* also varies according to a particular scenario. For example, an event coverage scenario may have a high density of nodes whereas a disaster recovery scenario might have a low density as the nodes are spread out over a wide area.

3) *Traffic rates* vary according to the node linkage failures, congestion and mobility. The sources and type of traffic (for example, CBR, TCP or UDP) must also be taken into account while defining the scenario. Normally, the traffic type used is Constant Bit Rate (CBR). The packet rate and size for a realistic scenario could be 4 packets/sec and 512 bytes respectively.

Sample scenarios and their respective parameters for simulations are listed in Table 2. Scenarios I and II are based on the *Reference Point Group Mobility model (RPGM)* [11]. RPGM is a group mobility model where each group has a logical center (similar to a troop head) that determines the group behavior. The nodes within a group move randomly according to the RWM, but overall the group movement is determined by the leader. Scenarios III and IV are based on entity mobility models. The most commonly used entity mobility model is the *Random Waypoint*. However, for realistic scenarios, in scenario III the *Manhattan Grid Model* is used, and in scenario IV the *Gauss Markov Model* is used.

Table 2: Sample Scenarios

parameters	<i>I. Battlefield</i>	<i>II. Rescue Operation</i>	<i>III. City traffic</i>	<i>IV. Event Coverage</i>
Mobility model	RPGM	RPGM	Manhattan Grid	Gauss Markov Model
Number of nodes	10 in each group 5 groups	5 in each group 10 groups	50	50
Area	2000 * 2000 m	1000 * 1000 m	1500 * 500 m	500 * 500 m
Speed	Node speed: 5 m/s Group speed : 1 m/s	Node speed: 2 m/s Group speed : 5 m/s	Node speed: 20 m/s	Node speed: 2 m/s Group speed : 5 m/s

5.2. Metrics

Having defined the parameters for the scenarios, we have identified the following metrics, based on which

the authentication mechanisms can be evaluated. Some of the metrics have been adapted from [4].

a) *Successful Certification Ratio* (μ) measures the ratio of the number of successful certification services (including issuance, NC_{ISS} , and renewal, NC_{REN} , respectively) to the total number of requests for such services ($NC_{TOT-ISS}$ and $NC_{TOT-REN}$, respectively). It gives an idea about the efficiency of the mechanism in providing successful certification services. If we consider μ_{REN} as the successful certification renewal ratio, and μ_{ISS} as the successful certificate issuance ratio, then their respective value can be calculated as follows:

$$\mu_{REN} = \frac{NC_{REN}}{NC_{TOT-REN}} \quad \mu_{ISS} = \frac{NC_{ISS}}{NC_{TOT-ISS}}$$

Here, NC_{REN} and NC_{ISS} are the respective total number of certificate renewed and issued, and $NC_{TOT-REN}$ and $NC_{TOT-ISS}$ the respective number of requests for certificate issuance and renewal.

b) *Settling time* (st) measures the initial time taken for all the nodes in the network to be issued valid certificates. The value of st can be calculated as the difference between the time when all the nodes are issued valid certificates and the starting time when the process of certificate issuance begins. The settling time taken will depend on factors such as the number of malicious or non-cooperative nodes, the algorithms used for key generation and distribution, etc. If the pre-authentication methods are efficient (R.5), the settling time will be less.

c) *Frequency of Certification* (f_{cert}) measures the number of certification services per time interval.

$$f_{cert} = \frac{N_{cert}}{T_{int}}$$

Here N_{cert} is the total number of certification services (issuance/renewal) by nodes in the network, and T_{int} is the simulation time. As the topology of the network changes, it is expected that there will be frequent certificate issuance and renewal processes. This incurs overhead, since each time a node wants to create or renew its certificate costly computations have to be carried out for the public key mechanism. We intuitively predict that a distributed and self-organized mechanism will have a lower frequency of certificate creation, renewal and revocation, and hence, a lower f_{cert} .

d) *Average Certification Delay* (acd) is measured as the time delay between the certificate service request ($CSReq$) and the certificate service reply ($CSRep$) averaged over the simulation time.

$$acd = \frac{\sum_{i=1..n}(CS Re p_i - CS Re q_i)}{T_{int}}$$

This value estimates the efficiency of the algorithm, and mainly depends on the time complexity of the algorithm.

6. Summary and Future Work

Successful authentication in mobile ad hoc networks are critical for assuring secure and effective operation of the supported application, especially in distributed field applications where mobile nodes are spread over a large geographical area. Several certificate-based authentication mechanisms have been proposed for MANETs. We survey some of these mechanisms, and charted out the requirements for certificate-based authentication schemes for MANETs. We also propose a few experimental scenarios and metrics, based on which simulation study of these methods are currently under way, using contiki OS with cooja simulations. [13].

References

- [1] Internet X.509 Public Key Infrastructure Certificate and CRL Profile - *RFC 2459*.
- [2] S. Capkun, L. Buttyan and J-P Hubaux. "Self-Organized Public-Key Management for Mobile Ad Hoc Networks ", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, Jan-Mar 2003, pp. 52-64
- [3] Weihong Wang, Ying Zhu, Baochun Li. "Self-Managed Heterogeneous Certification in Mobile Ad Hoc Networks ", in *the Proceedings of IEEE Vehicular Technology Conference (VTC 2003)*, Orlando, Florida, 10/6-9, 2003.
- [4] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. "Providing robust and Ubiquitous Security support for Mobile Ad Hoc Networks ", *Proceedings of the 9th International conference on Network Protocols (ICNP)*, Riverside, California, USA, November 11-14 2001.
- [5] Edith C. H. Ngai and Michael R. Lyu. "Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks", *24th International Conference on Distributed Computing Systems Workshops - W4: MDC (ICDCSW'04)*, Hachioji, Tokyo, Japan, 3/23-24, 2004.
- [6] L. Zhou and Z. Haas. "Securing Ad Hoc Networks", *IEEE Network magazine, special issue on networking security*, Vol. 13, No. 6, November/December 1999.
- [7] Matei Ciobanu Morogan, Sead Muftic. "Certificate Management in Ad Hoc Networks", *2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, January 27 - 31, 2003, pp. 337.
- [8] F. Stajano and R. J. Anderson. "The resurrecting duckling: Security issues for ad-hoc wireless networks" In *7th Security Protocols Workshop, volume 1796 of Lecture Notes in Computer Science, Cambridge, United Kingdom*, 1999. Springer-Verlag, Berlin Germany.
- [9] Dirk Balfanz, D. K. Smetters, Paul Stewart and H. Chi Wong: "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks", *Symposium on Network and Distributed*

Systems Security (NDSS'02), Xerox Palo Alto Research Center, Palo Alto, USA, 2002.

[10] P. Zimmerman. *The Official PGP Users guide*, MIT Press, 1995, ISBN 0-262-74017-6.

[11] T. Camp, J. Boleng, and V. Davies. "A Survey of Mobility Models for Ad Hoc Network Research", in *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, 2002.

[12] S. Selvakumar, Dr.S.Ravi, "Adaptive Modulation IN Reconfigurable Platform" *Journal of Theoretical and Applied Information Technology*. Vol 68, PP 108-114, Oct 2014.

[13] S. Selvakumar, Dr.S.Ravi, "DPSK and QAM Modulation Detection analyzed with BER Estimation" *IEEE International Conference on Current Trends in Engineering and Technology*, July 2014.

