View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Illinois Digital Environment for Access to Learning and Scholarship Repository

MOCA : MObile Certificate Authority for Wireless Ad Hoc Networks

Seung Yi Robin Kravets

Department of Computer Science
University of Illinois at Urbana-Champaign
201 North Goodwin Avenue, Urbana, IL 61801 USA

{*seungyi, rhk*}*@cs.uiuc.edu*

Report No. UIUCDCS-R-2004-2502, UILU-ENG-2004-1805

December, 2004

MOCA : MObile Certificate Authority for Wireless Ad Hoc Networks

Abstract

An authentication service is one of the the most fundamental building blocks for providing communication security. In this paper, we present the MOCA (MObile Certificate Authority) key management framework designed to provide authentication service for ad hoc wireless networks. MOCA is a distributed certificate authority (CA) based on threshold cryptography. We present a set of guidelines for a secure configuration of threshold cryptography to maintain strong security. MOCA utilizes a carefully selected set of mobile nodes to function as a collective certificate authority while the MOCA nodes are kept anonymous. Equipped with a novel routing protocol designed to support the unique communication pattern for certification traffic, MOCA achieves high availability without sacrificing security. Both the security of the framework and the operational performance is evaluated with rigorous analysis and extensive simulation study.

1 Introduction

A mobile wireless ad hoc network is formed by a group of mobile nodes with wireless communication capability without support from any stationary communication infrastructure. This unique infrastructure-less nature enables ad hoc networking technology to provide instant network deployment in situations where no communication infrastructure is available. Many proposed applications for ad hoc networks, including battlefield communication, disaster recovery and emergency rescue, involve mission critical situations where the security of the communication in the network must be guaranteed. Given these security requirements, the lack of adequate security support can easily negate the utility of an ad hoc network in practice.

Security begins with reliable authentication. Other security services, including access control, auditing, and authorization, all rely on an authentication service to provide reliable identification of participants. A well-established way to provide an authentication service in distributed systems is public key cryptography [24], where an entity is represented with a pair of keys. The public key is used as the ID of the entity while the private key is used to prove the ownership of the public key. The public key is disseminated in the form of a digital certificate that binds the entity's identity to the entity's public key. The successful use of public key cryptography requires an efficient mechanism to manage such digital certificates. One popular example is *Public Key Infrastructure (PKI)* [17]. PKI is usually designed around a centralized and trusted component called the *Certificate Authority (CA)*, which binds and unbinds entities to their public keys by issuing and revoking digital certificates, and also functions as the repository for active digital certificates.

However, it is questionable if the centralized CA-based PKI for wired networks is directly applicable to ad hoc networks, where different characteristics invalidate many assumptions that traditional PKI relies on. First, nodes in an ad hoc network are more vulnerable compared to wired hosts. Second, network topology and connectivity can rapidly change in ad hoc networks due to the mobility of nodes and the use of a wireless medium, making it difficult to maintain availability to a mobile CA. Therefore, an ad hoc key management framework must be designed to operate under these unique characteristics.

To address the challenges of ad hoc environments, the CA's functionality can be distributed to multiple nodes in such a manner that the CA stays secure even when some portion of the responsible nodes are compromised or become unavailable [29]. Most approaches in this direction rely on

a cryptographic technique called *threshold cryptography* [9]. However, there is an inherent tension between the two most important goals of a distributed CA: *strong security* and *high availability*. Careless focus on strong security can easily make the CA unavailable or too costly to be maintained. Similarly, a blind effort to increase the availability of a distributed CA can easily lead to a security breach of the CA. Therefore, an ad hoc key management framework must be designed as a comprehensive system addressing all the challenges in a unified framework.

The contribution of our work is the design and implementation of the *MOCA (MObile Certificate Authority)* ad hoc key management framework. MOCA uses threshold cryptography to distribute the CA functionality to multiple nodes. MOCA differs from other distributed CA approaches for ad hoc networks by limiting MOCA nodes to a small but more secure subset of the nodes in an ad hoc network. The distributed CA is then made available with novel communication support designed for the unique pattern that arises from the access to any threshold quorum system. These novel design characteristics enable MOCA to simultaneously provide strong security and high availability.

The rest of the paper is organized as follows. We first discuss the general challenge of ad hoc key management in Section 2. Section 3 describes the MOCA framework in detail. We then analyze the security of the MOCA framework with our novel security metric in Section 4. In Section 5, the efficiency of communication support in the MOCA framework is illustrated with an extensive simulation study and we conclude in Section 6.

2 Key Management in Ad Hoc Networks

The key role of a certificate authority is to act as the trust anchor for the system. As long as the CA is trusted, any certificates issued by the CA are universally trusted. Therefore, in any CA-based PKI, the security of the whole system relies on the security and availability of the CA. Traditionally in wired networks, a CA is deployed as a single host (or a cluster of hosts with a single front-end) that is powerful enough to handle all requests from the network and heavily guarded physically and electronically. Although the single CA can still be a bottleneck and also a single point of failure, it is relatively easy to protect a wired host from attacks and failures. There are many such CAs in operation for the Internet and they all appear as a single site to the rest of the Internet [8, 22, 23].

However, there are three significant differences that distinguish ad hoc networks from any other type of networks. First, nodes are assumed to be mobile and, therefore, lightweight and open to physical attacks. Second, the topology of an ad hoc network can be very unstable due to the mobility. Third, communication can only use the wireless medium, which is inherently lossy and error-prone. Infrastructure-based wireless networks like wireless LANs and cellular networks share the last challenge but the first two are unique to ad hoc networks. These differences call for a new way of providing a security service for an ad hoc environment.

2.1 Threat Model

The threats that can exist in an ad hoc network must drive the design of a key management framework for ad hoc networks. Attacks can be classified into two categories: *active* and *passive* . Active attacks involve behaviors such as manipulating packets, attacking other mobile nodes, or jamming the wireless medium. Passive attacks only rely on overhearing the traffic without disrupting network operation. Passive attacks are harder to detect compared to active attacks

with visible anomalies. We focus our attention on two active attacks on a distributed PKI: Routing layer attacks and Directed attacks on CA nodes.

- Routing Layer Attacks - Malicious nodes can disrupt routing behavior by advertising false routing information, injecting incorrect routing packets, or even luring all packets and dropping them [2, 10, 11, 12, 13, 16, 27]. Some routing layer attacks can be used to mount a simple denial-of-service attack if the attacker can either block or reroute all of the victim's packets. The MOCA framework uses a set of routing protocols based on the intelligent use of limited flooding that are immune to most routing layer attacks.

- Directed Attacks on CA nodes - When the attacker can discover either the identity or the location of CA nodes, the attacker can focus its resource in attacking only the CA nodes. The MOCA framework is designed to minimize the possibility of the CA nodes getting compromised. MOCA nodes are only selected from more secure and capable nodes and their identities are hidden so that an adversary cannot direct an attack on the MOCA nodes.

Radio frequency jamming is an active attack mounted at the physical layer of the network where an attacker transmits a high power signal over the spectrum, effectively *jamming* the band. This is a low-level denial-of-service attack and defense against it is out of scope of this paper.

Passive attacks include eavesdropping and traffic analysis. The MOCA framework is not vulnerable to eavesdropping since all information contained in communication between a client and the MOCA framework is public. On the other hand, traffic analysis may provide the attackers with sensitive information about the configuration of the framework. However, it is unclear how feasible a traffic analysis attack can be in ad hoc environments since traffic analysis requires a large amount of attackers' resources [15]. While there are known approaches to deter traffic analysis in wired networks [5, 6, 20], it is questionable if any of these approaches can be directly applied to ad hoc environments due to their excessive communication and computation overhead. Therefore, the design of our MOCA framework focuses on the two specified active attacks. Based on this threat model, we next present a set of requirements for effective ad hoc key management frameworks and examine existing approaches based on these criteria.

2.2 Three Goals for Ad Hoc Key Management

A successful key management framework for ad hoc networks must satisfy three fundamental requirements: fault tolerance, security, and availability. These terms are sometimes used interchangeably, mainly because they are not independent of each other. To avoid confusion, we first clearly define these terms.

- *Fault Tolerance*: The goal of fault tolerance is to maintain correct operation in the presence of faulty nodes. We restrict the definition of faulty to non-malicious. Fault tolerance is related to availability because as long is the service can tolerate the faults, such faults do not impact service availability.

- *Security*: Acting as the trust anchor for the whole network, a key management framework must be designed to be resilient against all levels of attacks and robust enough to withstand a relatively high fraction of compromised nodes.

- *Availability*: Traditionally, the term availability has been used in conjunction with fault tolerance. However, in ad hoc networks, availability is also highly dependent on network

connectivity. In wired networks, if the service is online, it is by definition available since connectivity between clients and the service is usually guaranteed. In ad hoc networks, clients may not be able to contact an operational service due to unstable and rapidly changing connectivity.

These three requirements are not independent of each other. For that reason, careless attempts to improve any one aspect may affect the others adversely. For example, a simple and very effective way to improve fault tolerance is to replicate the CA. However, this approach makes the overall framework more vulnerable. Interactions between these requirements must be throughly understood before designing a distributed key management framework. To understand these interactions, we discuss current approaches to providing PKI in ad hoc networks and how each approach tries to satisfy the three requirements.

2.3 Existing Approaches

The simplest approach to providing CA functionality in an ad hoc network is to assign a single node to be the CA. The success of this scheme depends on that single CA node. Since failure of one node breaks the entire system, fault tolerance is very low. Similarly, vulnerability is high since an adversary only needs to compromise one node to acquire the secret key. Finally, given the expected mobility and unpredictability of ad hoc networks, client nodes may not be able to reach the CA in a timely fashion, making availability very unpredictable. Therefore, it is clear that a single CA cannot effectively service a whole ad hoc network.

A higher fault tolerance can be achieved by replicating a fully functional CA on multiple nodes. With r replicas, the system can withstand *(r - 1)* failures because the CA is available as long as there is at least one operational CA. Availability is also improved since a client node has a better chance of reaching one of r CAs. Unfortunately, the security of the framework is now even lower than the single node CA scenario. An adversary needs only to compromise one of the r CA nodes to acquire the CA's secret key and so compromise the whole system. The problem of using replicated CAs stems from the fact that each replica has full knowledge of the system secret. Therefore, using replicated CAs is not applicable in ad hoc networks where nodes are more vulnerable.

Wu et al. first proposed the use of threshold cryptography to distribute a CA's functionality over multiple nodes for higher fault tolerance [25]. However, their approach is aimed at large-scale wired networks like the Internet and is not directly applicable to ad hoc environments. Zhou and Haas adapted Wu's approach into ad hoc networks [29], stating the problem clearly but only presenting the conceptual design at a very high level. While Zhou and Haas laid out the basic concepts for an ad hoc distributed CA, they missed the important aspects of the availability of the distributed CA in the ad hoc network and the communication overhead from accessing the CA.

Kong et al. address the availability issue by making every node in the network share CA functionality [18]. A client only needs to contact any k nodes to get a certification service. Assuming there are more than k nodes in a client's one hop neighborhood, the client can get a certification service cheaply by using a single one-hop broadcast for the request. While this solution addresses availability and fault tolerance, it compromises the security of the system. In general, the gap between k and n in secret sharing schemes defines the security of the system. k can be anywhere between 1 and n. As k approaches n, thus shrinking the gap between k and n, the system becomes more secure because an adversary needs to compromise more nodes to penetrate the system. However, if k is too large, the system becomes less available to clients and also less tolerant to faults. When k approaches 1, increasing the gap between k and n, the effect is reversed and the

system becomes more available but less secure. Kong chose to keep k relatively small to address the availability problem and ended up with a vulnerable system where any adversary only needs to compromise a small fraction of nodes in the network to collapse the service.

Another notable scheme is proposed by Hubaux et al. [14], which is later extended in [3]. In their scheme, every node acts as its own CA, similar to the PGP [30] "Web of Trust" model. The main difference between PGP and their scheme is that there is no longer a well-known certificate directory where all certificates are stored. Instead, every node in the network carries a part of the certificate directory. In PGP, when two users wish to authenticate each other, they must search the certificate directory for a chain of certificates that links both users. In Hubaux's scheme, this problem is transformed into finding an intersecting point between the certificate chains carried by each user. While their approach is practical for purely self-organized networks, it cannot provide a high assurance authentication service as provided by the certificate authority in PKI.

3 MOCA

In this section, we present the MOCA (MObile Certificate Authority) ad hoc key management framework. MOCA uses threshold cryptography to divide and distribute the CA functionality to multiple nodes. MOCA nodes are picked carefully based on their characteristics such as physical security, computational capability, and trustworthiness. Based on these criteria, a relatively small set of nodes is selected to function as the distributed CA. These MOCA nodes operate in the network without revealing their identity as the CA nodes. Careful MOCA node selection and anonymity help achieve a high level of security for the MOCA framework. Usually, the price for having a small set of CA nodes is reduced availability and higher communication overhead. MOCA addresses this problem by employing a suite of specially designed routing protocols for efficient communication support for the certification traffic.

3.1 Using Threshold Cryptography

MOCA employs *k-out-of-n* threshold cryptosystem to divide the CA's master private key to n distributed CA nodes [9, 21]. The CA's master private key is divided to n pieces and any k of such pieces can be used to reconstruct the master private key. There are three important factors to consider when employing threshold cryptography: (1) Who will be given a secret share? (2) Who will distribute the secret shares? And, (3) How to handle compromised secret share holders? Next, we discuss each of these questions in detail.

3.1.1 Choice of MOCA Nodes

Any number of nodes between one and the total number of nodes in the network can be selected as MOCA nodes. However, selecting a relatively small fraction of nodes in the network is desirable. Intuitively, the crypto threshold k cannot be set too high since it causes every certification request to generate excessive communication overhead. With the crypto threshold value fixed, increasing the number of MOCA nodes widens the gap between k and n, allowing attackers to more easily locate enough MOCA nodes to compromise and steal the master private key. Therefore, n must be kept to a relatively small value compared to the total number of nodes in the network.

There are many possible ways to select the MOCA nodes. While most research in ad hoc networking has implicitly treated all nodes to be identical, it is more likely that an ad hoc network

contains several types of mobile nodes that are different from one another in power capacity, transmission range, computational capacity, and security. Therefore, any security service or framework should utilize this potential *heterogeneity*. For example, consider a battlefield scenario with a battle group consisting of infantry soldiers, platoon commanders' jeeps, company commanders' command vehicles, artillery vehicles, transport vehicles, and tanks. All of these mobile nodes have different rank, power, computation capacity, transmission range, and level of physical security. In such a case, it would be wise to pick nodes with higher ranks, more power, more capabilities and stronger security to provide a security service. While it may not be necessary to exploit this potential heterogeneity to enhance basic ad hoc routing, certainly this heterogeneity can be used to help improve network security by endowing *more secure* nodes with sensitive information. Similar situations can be imagined in emergency rescue operations, disaster recovery, or any other scenarios where ad hoc networks can play a critical role. In general, knowledge of such heterogeneity should be used to determine the nodes that share the responsibility of the CA. Once the number of MOCA nodes, n, is chosen, the MOCA framework selects n mobile nodes with (1) highest trustworthiness, (2) highest physical security, (3) most computational capacity, and (4) most power as MOCA nodes.

Once the MOCA nodes are selected, their identity must be kept secret. Maintaining the anonymity of MOCA nodes is crucial to achieve strong security. Intuitively, it is harder, if not impossible, for an adversary to locate anonymous MOCA nodes and compromise them. This forces the adversary to invest far more resources trying to compromise the key management framework and reduces the chance for successful attacks. With routing layer support for anonymous communication and careful design of protocol message contents, the MOCA framework provides certification service anonymously.

3.1.2 Off-line Key Dealer

Once a set of nodes is selected to serve as MOCA nodes, the nodes are configured by an off-line key dealer that does not participate in the ad hoc network operation. The off-line key dealer generates the CA's master key pair, divides the private key using threshold cryptography, distributes the key shares to selected MOCA nodes and then goes offline and stays out of the network. This off-line key dealer is a crucial component to the framework's overall security since it holds the full secret key of the CA. Therefore, it is critical to protect the off-line key dealer from attackers.

Alternative approaches that do not require such an off-line key dealer use online election of CA nodes. Such online election can be performed when more than a threshold number of CA nodes become disabled or there is a network configuration change. Kong et al. uses online election to make the distributed CA self-contained within an ad hoc network [18]. In their approach, a group of neighboring nodes can promote another node to become a CA node with a proper key share. However, this design decision opens the door for a serious security breach since it is relatively easy to compromise enough nodes to steer the online CA election to benefit attackers. Also, given the problem of Sybil attacks, it is impossible to prevent impersonation in a open distributed system without clearly distinguished centralized authentication support [7]. Therefore, Kong et al.'s proposal is also open to Sybil attacks where an adversary node may acquire multiple identities to impersonate enough nodes to acquire key shares and reconstruct the CA's full private key, resulting in the total compromise of the framework. Defense against Sybil attacks in ad hoc networks is still an open research problem [19]. Therefore, MOCA does not allow online election of new CA nodes in the interest of maintaining strong security.

3.2 Message Format

Hiding the identity of the MOCA nodes is another important factor for strong security. It is also crucial that no message exchanged between a client and the MOCA nodes carry any identifying information about the MOCA nodes. There are two types of messages that can be exchanged between the MOCA framework and client nodes: certification requests from a client and certification replies to a client. All messages are designed so that the identities of MOCA nodes can stay hidden.

3.2.1 Certification Request (CREQ)

A certification request message is sent from a client node to a group of MOCA nodes. All MOCA nodes that receive the request message must reply accordingly. A certification request message contains the following information.

1. Client ID - the ID of the client node
2. Type - A certificate request, a revocation request, or a certificate retrieval request.
3. Payload - A certification request contains the public key of the requesting node. A revocation request contains the public key of the requesting node. A certificate retrieval request contains the ID of a node whose certificate is being requested.
4. Signature - The whole certification request message is signed with the requesting node's private key. Note that this key may not yet be certified at this point.

3.2.2 Certification Reply (CREP)

When a MOCA node receives a certification request message from a client, it returns a certification reply message containing the following information.

1. Client ID - The ID of the requesting node
2. Payload - For a certification request, a partial signature over the digital certificate is sent back. For a revocation request, a revocation certificate is created and sent back. For a certificate retrieval request, a digital certificate is sent back if available at the MOCA node.

Note that a CREP message does not contain any information about the replying MOCA node to keep them anonymous.

3.3 Communication Support for the Certification Traffic

The design decision to limit the number of MOCA nodes in a network creates a problem for availability. Since there are a limited number of MOCA nodes to choose from, a client node may have difficulty contacting enough MOCA nodes. Without any attention, this can also easily put excessive communication overhead on the network while causing network-wide congestion and wasting the scarce resources of mobile nodes. The key idea of our solution lies in the observation of a novel communication pattern generated by the MOCA framework. When a client contacts a set of MOCA nodes for a certification service, it generates a communication pattern of *one-to-many-to-one*. The client needs to contact k MOCA nodes and must receive k independent replies. We named

this communication pattern *manycast* and performed an extensive study of its characteristics [4]. This transactional manycast communication pattern may look similar to multicast but there are several differences. First, multicast can only handle the first half of a manycast transaction. The *many-to-one* part cannot be managed by a multicast protocol. Second, in multicast, a source wishes to contact *all* members of the multicast group whereas in manycast, the source wishes to contact a fixed size subset of server nodes.

When designing a protocol to support manycast communication, there are two questions to answer: (1) How to choose and contact k nodes from the set of n without knowing their individual identities? And (2) What is the most efficient way to contact and receive individual replies from them? Among the suite of manycast routing protocols proposed in [4], only two are appropriate for the MOCA framework: *flooding* and *scoped-flooding*, since only these two approaches can maintain the servers' anonymity.

A client without any knowledge about the network or MOCA nodes must flood a certification request (CREQ) the first time. Every MOCA node that receives the CREQ replies with a certification reply (CREP) generated with their secret share. A CREP message is unicast back to the client using the reverse path created by the CREQ message. When a client receives CREP messages, the client records the hopcount of each reply. These hopcounts show the distance between the client and the MOCA node that sent the reply messages. When the client needs to send the next CREQ message, the information in this hopcount cache is first examined to find out whether it is possible to reduce the scope of flooding controlled by the TTL (Time-To-Live) field in the CREQ message while reaching enough MOCA nodes. With scoped flooding, the MOCA framework achieves highly efficient communication support for certification traffic while keeping the MOCA nodes secure and anonymous. A detailed performance study of manycast routing in the MOCA framework is presented in Section 5.

4 Security Analysis

In this section, we examine the security of the MOCA framework. Threshold cryptography used to distribute the CA functionality has built-in support for security and fault tolerance. More specifically, an adversary must compromise at least k MOCA nodes to compromise the CA. As long as there are k non-faulty MOCA nodes in operation, the framework can provide service. However, deployment of a distributed CA with threshold cryptography requires careful attention to finer details. We first examine the basic parameters and their relationship to the security of the framework and then discuss some additional precautions required for secure deployment.

4.1 Threshold Cryptography Parameters

The configuration of the MOCA framework is determined by the total number of nodes in the network, M, the crypto threshold for secret reconstruction, k, and the number of MOCA nodes, n. While M cannot be chosen *a priori*, the crypto threshold, k, can be selected and it is important to understand the effects of the selected value of k. k can be chosen between 1 (a single MOCA node must be contacted for a certification service) and n (all MOCA nodes must be contacted for a certification service). Setting k to a higher value has the effect of making the system more secure since k is the number of MOCA nodes an adversary needs to compromise to penetrate the system. But at the same time, a higher k value makes clients contact more MOCA nodes for certification service, which may result in higher communication overhead. Therefore, the choice of k must strike

a balance between the two conflicting goals by being small enough to not overwhelm the network but large enough to withstand attacks.

The number of MOCA nodes, n, is determined by the characteristics of the nodes in the network and is determined before the MOCA framework is deployed. n can be changed to a new value but it requires costly intervention of the off-line key dealer. In a threshold system, n defines the limits of the system as an upper bound for k since $1 \leq k \leq n$. Given a fixed value of k, a larger n increases the availability of the whole framework since a client can choose from a larger set of MOCA nodes. On the other hand, *(n - k)* is the maximum number of faults the framework can survive, so a larger n also means higher fault tolerance.

4.2 Measuring the Security Level for Distributed CAs

Assuming the distributed CA nodes are anonymous and an adversary cannot discover their identity, the best approach for the adversary is to compromise as many nodes as possible in a given amount of time, hoping that enough CA nodes are included among the compromised nodes. The following simple combinatoric equation captures this situation.

$$\text{Security Level} = 1.0 - \frac{\sum_{i=k}^{c} \binom{n}{i}\binom{M-n}{c-i}}{\binom{M}{c}},$$

This formula measures the probability that an attacker fails to compromise the distributed CA given that the attacker can compromise at most c nodes in a window of time. If an attacker is capable of pinpointing attacks only on the CA nodes, the attack always succeeds as long as $c \geq k$. Therefore, it is crucial to keep the CA nodes anonymous, limiting attackers to random attacks.

Currently, there are only two concrete designs proposed for distributed CAs [18, 26]. In Kong's approach, every node serves as a CA node. Therefore, it is impossible to hide the identities of CA nodes and application of this metric always yields a zero security level as long as $c \geq k$. In contrast, MOCA hides the CA nodes' identities as well as limits their number. Therefore, the best chance an adversary has to compromise the system is by randomly compromising as many nodes as possible. For example, the security level of a 30-node MOCA framework with $k = 5$ and $c = 10$ in a 150-node network is calculated to 0.97, which shows that it is not very likely that this configuration of the MOCA framework will be compromised.

This metric also reveals another important configuration pitfall that can be easily overlooked: the meaning of the gap between k and n. Based on the discussions so far, a large enough k and much larger n appear to be the right choice for configuration parameters. However, if the gap between k and n is too big, the security of the framework degrades. If $c < k$, the framework is secure by design since no adversary can compromise enough MOCA nodes. However, if $c \geq k$, it is possible for an adversary to compromise the framework. Therefore, it helps to limit the MOCA nodes to be a small set of more secure and capable nodes, which makes it harder for the adversary to locate and compromise enough MOCA nodes. Figure 1 (a) illustrates one example. Out of 150 total nodes in the network, 10, 30, 50, 100, and 120 nodes are selected as MOCA nodes with a fixed crypto threshold $k = 10$. The five curves in the graph display cases of $n = 120$, 100, 50, 30, and 10 from left to right. As the attacker's capacity c increases, all curves monotonically decrease from 1.0 to 0.0. However, the rate of decrease is much higher for curves with a larger n. This illustrates the effect of the gap between k and n, which shows that a too large n can weaken the configuration of the overall framework.

4.3 Selection of MOCA nodes

As discussed in the previous section, the number of MOCA nodes, n, in a network should be limited to a reasonable number. It may seem counterintuitive to limit the number of MOCA nodes, which may reduce the availability and the fault tolerance achieved by the distributed nature of MOCA. For example, in a 300 node network, an operator may have a choice of selecting 200 random nodes or 30 nodes with higher physical security to support CA functionality. Blindly comparing the number of MOCA nodes in the system, the first choice seems better because it has more MOCA nodes in the network, improving fault tolerance and availability. However, by guaranteeing a higher level of security of the 30 MOCA nodes in the second case, compromising them becomes much harder than compromising the randomly selected MOCA nodes in the first case, hence making the second case more secure against adversaries. It is possible that an ad hoc network does not have enough heterogeneity among the nodes, which may make it difficult, if not impossible, to choose MOCA nodes based on heterogeneity. In such cases, we can fall back to random selection of MOCA nodes. However, the level of security will decrease since there is no guarantee on the security of each MOCA node.

The next question is how to pick the best subset of nodes to serve as the MOCA nodes. As discussed in Section 3.1.1, MOCA nodes are selected based on their characteristics and limited to more secure, more capable, and more trustworthy nodes. This design decision makes it more difficult for an adversary to compromise the MOCA framework, in effect decreasing the adversary's attack capacity c. Figure 1 (b) illustrates an example. The three curves display the security levels for the configurations with $c = 50$, 30, and 20. The horizontal line at the security level 1.0 is for the case of $c = 5$, where the framework is completely secure since $c < k$. The curves move to the right as the adversary's attack capacity is decreased, making the system more secure. A careful selection of MOCA nodes can indeed help maintain a higher security level.

5 Communication Performance Evaluation

The focus of our performance evaluation is to measure the effectiveness and efficiency of MOCA communication support. We show that the MOCA framework can maintain a secure distributed CA without incurring prohibitive communication overhead by employing scoped-flooding. Effectiveness while measured by the success ratio of certification requests. Given a crypto threshold k, more than k replies from MOCA nodes makes a certification request successful. The success ratio must be kept at a high level under all circumstances to provide useful service. However, a high success ratio should not come at the price of excessive overhead that can affect normal network operation. Overhead is measured by the number of messages transmitted per certification request. The simulation results show that scoped-flooding achieves a very high success ratio comparable to pure flooding with an acceptable packet overhead.

5.1 Simulation Set-Up

We implement our certification protocols in the ns-2 network simulator [1]. 150 mobile nodes are set up within either 1km by 1km area or 2km by 2km area. Out of 150 nodes, 30 nodes are randomly selected as MOCA nodes. 30 MOCA nodes represent 20% of the total nodes, which we believe provide a reasonable number of MOCA nodes to support the given ad hoc network. Each simulation is run for 600 seconds. Detailed simulation parameters are listed in the Table 6. The certification request pattern includes 100 non-MOCA nodes, each making 10 certification

requests randomly distributed through the simulation timeline, for a total of 1000 certification requests. Each requesting node makes one request per minute on average during the course of the simulation. This is roughly 100 requests per minute and we believe that this is a reasonable number if not too stressful to the framework. Assuming each certification request precedes initiation of a new secure communication, starting one secure communication session per node per minute should be more than adequate for ordinary mobile nodes. Node movement follows the random waypoint mobility model implemented by Yoon et al. [28]. Data points in the graphs are averaged over five different mobility scenarios with identical simulation parameters. We measure the performance of scoped-flooding with pure flooding as the baseline since they are the only anonymous manycast routing protocols available. Pure flooding always floods the network with certification requests, potentially incurring high overhead. In comparison, scoped-flooding uses a limited flooding of certification requests with a reduced TTL value when there is enough cached information. Scoped-flooding only falls back to pure flooding when there is not enough information in the hopcount cache. For all simulations, we control two parameters that affect the performance of the MOCA framework: the crypto threshold k and the mobility of nodes as measured in maximum speed.

- Crypto Threshold k - k is the minimum number of CREPs required for a client to reconstruct the MOCA's full signature and render the certification request successful. If k is set to a small number, a client only needs to collect a small number of k partial signatures to continue. Therefore, with a small k, the success ratio increases and the packet overhead decreases. A large k value makes attacks more difficult, but the burden on clients and the packet overhead increase since a client needs to contact a large number of MOCA nodes for a certification request.

- Mobility (Maximum Speed) - As nodes move faster, it becomes harder to maintain connectivity to enough MOCA nodes. When scoped-flooding is used, the client relies on its previous knowledge about the number of nearby MOCA nodes. Under high mobility, this knowledge remains valid only for a short period and scoped flooding fails more frequently, resulting in decreased success ratio.

Our simulation results show consistent patterns throughout different pause times, speed patterns and number of MOCA nodes. Therefore in this section, we only present the results for varying k with a fixed maximum speed of 20 m/s and varying maximum speed with a fixed $k = 15$.

5.2 Success Ratio

Success ratio for pure flooding stays higher than 98% under all crypto threshold values in the 1km by 1km scenario, showing the effectiveness of flooding as a manycast communication protocol (Figure 2 (a)). Scoped-flooding also maintains a high success ratio between 96% and 99%. The success ratio for scoped flooding is at its lowest with $k = 15$ when many scoped-flooding attempts fail because of the stale information in the hopcount cache. Success ratio again increases as k grows larger than 15 because there is not enough information cached at the client and the client falls back to pure flooding. A similar pattern is amplified in Figure 3 (a) since the larger area and the lower node density affect the success ratio adversely.

Both flooding and scoped-flooding are not affected by the mobility until the maximum node speed reaches 20 m/s. With mobile nodes traveling faster than 20 m/s, the success ratio of scoped-flooding degrades down to 52% under an extreme maximum speed of 50 m/s (Figure 2 (b)). This shows the effect of having stale information in the client's cache due to high mobility, which results

in more failed scoped-flooding attempts. Figure 3 (b) shows a similar pattern in the 2km by 2km scenario.

The effectiveness of the MOCA framework is demonstrated with these results. MOCA is capable of providing almost perfect availability under reasonable network conditions and the performance gradually degrades as the network condition becomes pathological.

5.3 Packet Overhead

To measure the packet overhead for the MOCA framework, we measure the number of total packets transmitted per request. When a packet is broadcast, it is counted once per hop. Unicast packets are counted at each hop. While the number of packets per request simply measures the amount of packet overhead, the number of packets per *satisfied* request shows the effect of the success ratio.

In Figure 4 (a), both flooding and scoped flooding show little difference under all k values in the 1km by 1km scenario. For both cases, the number of packets per *satisfied* request is a little higher than the number of packets per *all* requests since only a small portion of certification requests fail. However, Figure 5 (a) shows the power of scoped flooding in a larger area. The packet overhead of scoped-flooding is less than half of pure flooding in the lower k range. This shows the effect of localizing manycast transactions by scoped-flooding, which improves the scalability of the overall framework. As scoped flooding fails more often with k larger than 15, the overhead catches up to pure flooding.

While flooding shows a similar result under varying mobility in Figure 4 (b), the number of packets per satisfied request for scoped-flooding increases to higher than 500 packets per satisfied request. This shows the limitation of scoped-flooding to cope with very high mobility. Figure 5 (b) displays a similar pattern in a larger area. Again, scoped-flooding performs better in the lower mobility range but the overhead per satisfied request grows very quickly as mobility increases.

These results show that MOCA's scoped-flooding can effectively suppress the packet overhead by limiting the certification traffic to local regions. In small scale scenarios with 1km x 1km area, scoped-flooding incurs a little lower overhead than pure flooding. However, as the network size grows, scoped-flooding successfully suppress the overhead explosion from pure flooding. Scoped-flooding is a highly efficient and scalable approach to provide manycast communication support.

6 Conclusions and Future Work

In this paper, we present MOCA, a practical key management framework for ad hoc wireless networks. We clarify the necessity and the challenge of providing a PKI framework for ad hoc networks and identify the requirements for such a framework. Based on our observation of the potential heterogeneity among mobile nodes, we provide an intelligent way to pick a set of CA nodes. These selected secure nodes are called MOCA nodes and share the responsibility of collectively providing the CA functionality for an ad hoc network without revealing their identity. To achieve both strong security and high availability of the MOCA framework, we provide insight into the secure configuration of threshold cryptography and the observation of a novel communication pattern named *manycast*. To minimize the usage of scarce resources in mobile nodes, we develop a set of efficient and effective manycast communication protocols for mobile nodes to correspond with the MOCA framework. Our security analysis shows that the MOCA framework can be configured to defend against capable attackers and our simulation results show the effectiveness of manycast communication support.

There are still several interesting questions to be investigated. It is unclear if the MOCA framework is indeed vulnerable to a traffic analysis attack. We plan to study the types of information that can be inferred from certification traffic patterns and possible defenses against such attacks. Also, in this paper, we looked at the scalability issue related to packet overhead. We plan to perform a more detailed study on the scalability of manycast routing protocols under different node density, varying network size under a spectrum of network conditions.

References

[1] The ns-2 Network Simulator. Available at http://www.isi.edu/nsnam/ns/.

[2] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the ACM workshop on Wireless security*, 2002.

[3] S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility helps security in ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, June 2003.

[4] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets. Manycast: Exploring the space between anycast and multicast in ad hoc networks. In *Proceedings of the Ninth ACM Annual International Conference on Mobile Computing and Networking (Mobicom 03)*, September 2003.

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), February 1981.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[7] J. Douceur. The sybil attack. In *Proceedings of IPTPS 02 Workshop*, March 2000.

[8] Entrust. Entrust, Inc. Company homepage available at http://www.entrust.com/.

[9] Y. Frankel and Y. G. Desmedt. Parallel Reliable Threshold Multisignature. Technical Report TR-92-04-02, Univ. of Wisconsin–Milwaukee, 1992.

[10] Y.-C. Hu, D. B. Johnson, and A. Perrig. Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, June 2002.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002)*, September 2002.

[12] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocom 2003*, April 2003.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the the Second ACM workshop on Wireless security (WiSe 03)*, 2003.

[14] J. P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, 2001.

[15] S. Jiang, N. Vaidya, and W. Zhao. Routing in packet radio networks to prevent traffic analysis. In *Proceedings of the IEEE Information Assurance and Security Workshop*, July 2000.

[16] K. Sanzgiri and B. Dahill and B. Levine and C. Shields and E. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP 2002)*, November 2002.

[17] S. Kent and T. Polk. Public-key infrastructure (x.509) (pkix) charter. Available at http://www.ietf.org/html.charters/pkix-charter.html.

[18] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Proceedings of the 9th IEEE International Conference on Network Protocols (ICNP 2001)*, 2001.

[19] J. Newsome, R. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: Analysis and defenses. In *Proceedings of IEEE International Conference on Information Processing in Sensor Networks (IPSN 04)*, April 2004.

[20] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[21] V. Shoup. Practical threshold signatures. *Lecture Notes in Computer Science*, 1807, 2000.

[22] Thawte. Thawte, inc. Company homepage available at http://www.thawte.com.

[23] Verisign. VeriSign, Inc. Company homepage available at http://www.verisign.com/.

[24] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 1976.

[25] T. Wu, M. Malkin, and D. Boneh. Building intrusion tolerant applications. In *Proceedings of the 8th USENIX Security Symposium*, pages 79–91, 1999.

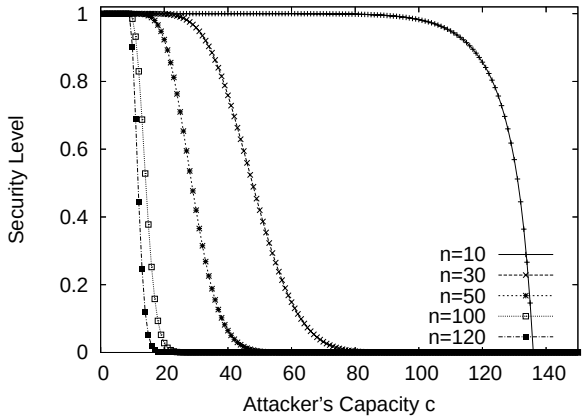
[26] S. Yi and R. Kravets. MOCA: Mobile certificate authority for wireless ad hoc networks. In *Proceedings of the 2nd Annual PKI Research Workshop (PKI 03)*, April 2003.

[27] S. Yi, P. Naldurg, and R. Kravets. Integrating quality of protection into ad hoc routing protocols. In *Proceedings of The 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 02)*, August 2002.

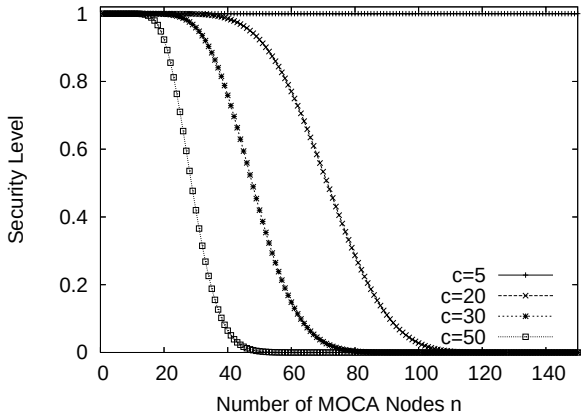
[28] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 205–216. ACM Press, 2003.

[29] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. *IEEE Network Magazine*, November 1999.

[30] P. Zimmermann. The official PGP user's guide. MIT Press, 1995.



(a) Varying the Number of MOCA Nodes n (k=10)

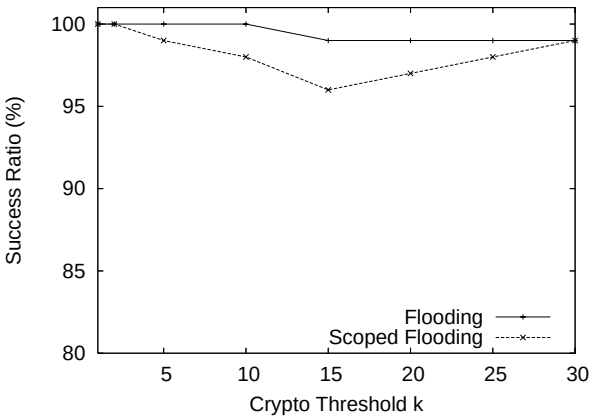


(b) Varying the Attacker Capacity c (k=10)

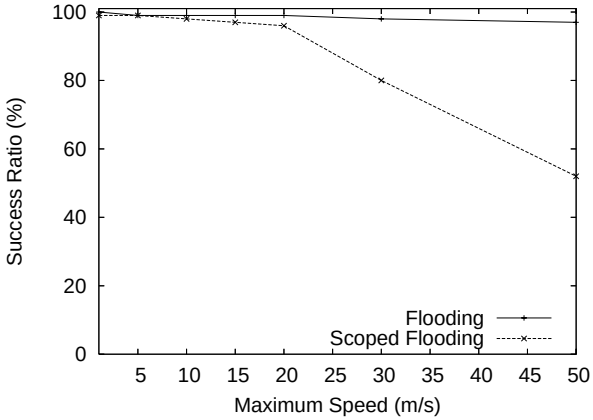
Figure 1: Security Level

Total Number of Mobile Nodes	150
Number of MOCA nodes	30
Area of Network	1000m x 1000m, 2000m x 2000m
Total Simulation Time	600 sec.
Number of Certification Requests	10 requests each from 100 non-MOCA nodes
Node Pause Time	0, 10 sec.
Maximum Node Speed	0, 1, 5, 10, 15, 20, 30, 50 m/s
Crypto Threshold k	1, 2, 5, 10, 15, 20, 25, 30

Table 1: Simulation Parameters

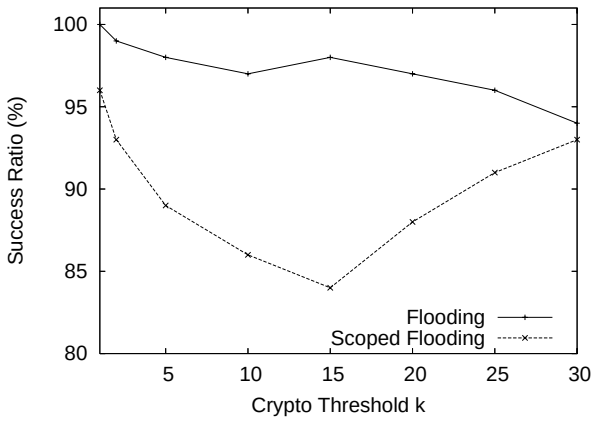


(a) Varying k (Max. Speed 20 m/s)

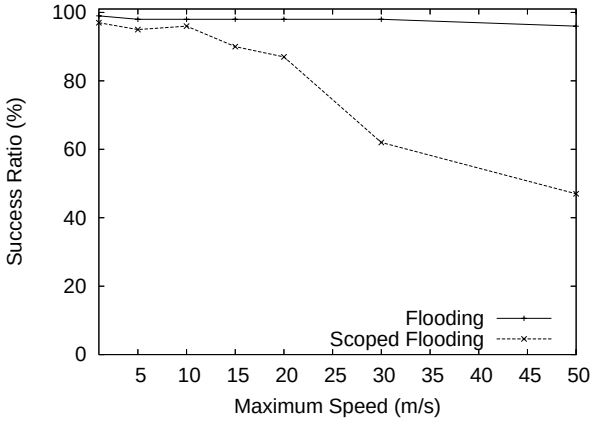


(b) Varying Mobility ($k = 15$)

Figure 2: Success Ratio in the 1000m x 1000m Scenario

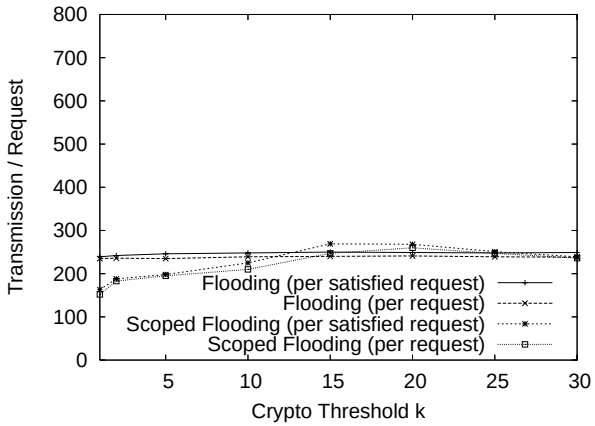


(a) Varying k (Max. Speed 20 m/s)

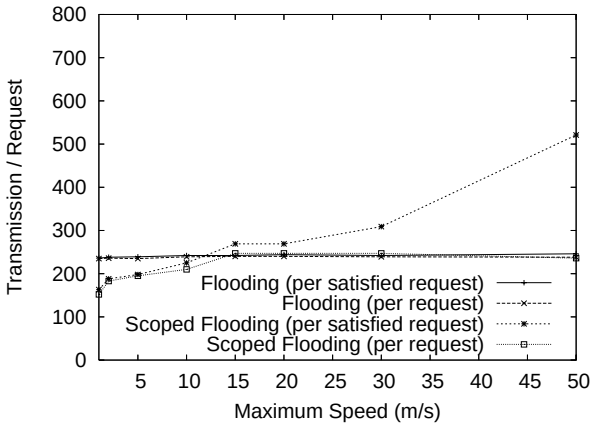


(b) Varying Mobility ($k = 15$)

Figure 3: Success Ratio in the 2000m x 2000m Scenario

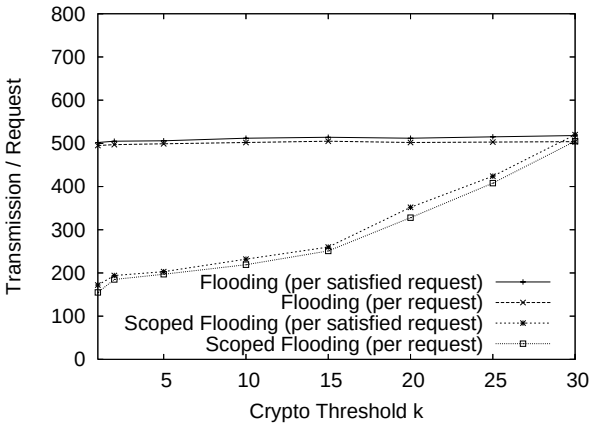


(a) Varying k (Max. Speed 20 m/s)

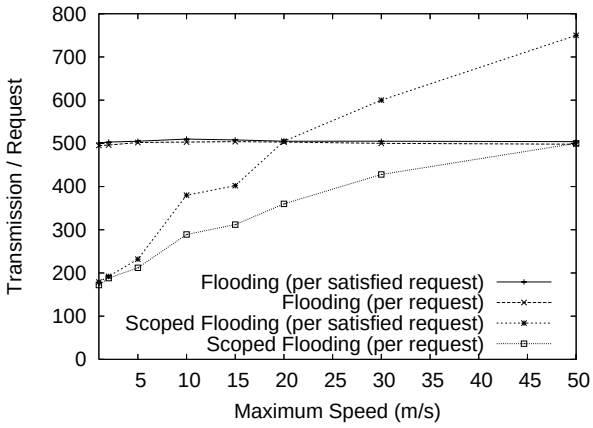


(b) Varying Mobility ($k = 15$)

Figure 4: Packet Overhead in the 1000m x 1000m Scenario



(a) Varying k (Max. Speed 20 m/s)



(b) Varying Mobility ($k = 15$)

Figure 5: Packet Overhead in the 2000m x 2000m Scenario