Composite Key Management for Ad Hoc Networks

Seung Yi   Robin Kravets
Department of Computer Science
University of Illinois at Urbana-Champaign
{seungyi,rhk}@cs.uiuc.edu

Abstract

In this paper, we present Composite Key Management, a novel paradigm for key management in ad hoc networks. While most existing approaches try to fit techniques developed for wired environments into ad hoc networks, our approach works within the specific limitations of ad hoc networks to increase security and availability of the key management framework. By composing techniques from PKI and certificate chaining, composite key management follows two fundamental principles that must be satisfied by an ad hoc key management framework: node participation and the use of a trusted third party. We introduce enhanced metrics of authentication to aid end users in understanding and using the key management framework. Through simulation studies, we demonstrate the effectiveness of composite key management under stressful scenarios where previous approaches fail and show that composite key management can address the challenges posed by the unique nature of ad hoc networks.

1. Introduction

Many target applications for ad hoc networks require strong communication security to operate. Examples include battlefield communication support and emergency rescue operations. However, the same infrastructureless nature of ad hoc networks that is good for easy, fast, and cost-effective deployment makes it difficult to support secure communication. Many security solutions rely on public key cryptography, the deployment of which requires the effective management of digital certificates through two fundamental services: secure binding of a cryptographic key to an entity (e.g., a user, a mobile node, or a service) and validation of such bindings to other entities. Most key management frameworks and other security services designed for wired networks and infrastructure-based wireless networks rely on a trusted infrastructure for security-related functions. However, in an ad hoc network without any infrastructure support, most traditional solutions are not directly applicable. The goal of our research is to understand the specific challenges for providing key management in ad hoc networks and use this understanding to design an effective key management framework.

To be effective, an ad hoc key management framework must satisfy three requirements. First, the operations of the key management framework must be secure against malicious attacks. While it is simple to guard wired nodes from physical attacks, many if not all of the nodes in ad hoc networks are expected to be mobile hosts that are more vulnerable to physical attacks. Therefore, a key management framework for ad hoc networks must be resilient to a higher fraction of compromised nodes. Second, the framework must be robust against non-malicious faults and still be available to the network. In other words, a high level of fault tolerance is necessary. Third, while wired networks assume persistent connectivity and so availability, ad hoc networks are expected to have frequent topology changes and even temporary disconnections. Therefore, a key management framework for ad hoc networks must be able to compensate for periodic disruptions in connectivity.

Similar to approaches targeted at wired networks, current approaches to ad hoc key management mainly utilize one of two key techniques: (1) public key infrastructure (PKI) [6] using a distributed certificate authority and (2) self-organized certificate chaining. To adapt PKI to ad hoc networks, threshold cryptography is used to provide a virtual certificate authority (CA) comprised of multiple mobile nodes that collectively provide certification services [8, 13, 15]. Since the virtual CA plays the key role of trust anchor for the rest of the network, it must be kept secure and reliable at all times. At the same time, the virtual CA must be efficient in computation and communication to be feasible in ad hoc networks. While virtual CA approaches can provide higher levels of assurance and require no warm-up period, they usually impose higher maintenance overhead.

Certificate chaining fits naturally with ad hoc networks where there is no physical infrastructure, relying on each mobile node to issue certificates to other nodes at their own discretion. Certificate chaining does not require any bootstrapping of the system, which fits well with self-organizing nature of ad hoc networks. However, certificate chaining requires a warm-up period to populate the certification graph, which completely depends on the individual node's behavior and mobility. Additionally, there are no guarantees that the resulting certification graph will be dense enough to be useful. Finally, the validity of a certificate chain depends on the trustworthiness of all the mobile nodes in the chain, which may not be easy to ensure in open networks. This dependence on potentially unknown nodes and the lack of any trust anchor in the system make certificate chaining unsuitable for situations requiring strong security guarantees.

While both approaches have different advantages and limitations, neither approach is effective in all scenarios. To address their limitations in context, we define two underlying principles for providing secure key management in ad hoc networks. First, the burden of key management should be distributed to all nodes. Essentially, the more nodes participating in key management, the more available the framework. However, it is important to distribute key management functionality in a way that maintains a high level of security. Second, it is highly beneficial to provide a trusted third party as a trust anchor for the network. Without a trust anchor, the amount of confidence in authentication cannot exceed a certain level. The presence of a trusted third party can significantly increase the confidence in authentications.

To take advantages of the benefits of both techniques and satisfy these principles, we propose *Composite Key Management*, which simultaneously deploys multiple key management mechanisms, including virtual CAs and certificate chaining. By combining the characteristics of both of these mechanisms, composite key management can provide high quality authentications with a high level of security and almost ubiquitous availability. A composite key management framework can also adapt to dynamic changes in the availability of key management services. For example, composite key management can provide excellent service in a network that supports both a virtual CA and certificate chaining. However, if one of the services is not available to a node, the node can still use the remaining services to receive the best possible authentication service. Essentially, users have a full spectrum of choices in how to participate in and use the service.

To complete our framework, we present an authentication metric to determine the trust level of the authentication and evaluate it as compared to pure virtual CA or pure certificate chaining frameworks. These evaluations demonstrate that the composite key management can be used to augment both virtual CAs and certificate chaining, improving both the success ratio for authentication and the level of confidence in the authentication.

The rest of the paper is organized as follows. Section 2 presents two principles for ad hoc key management, evaluates existing approaches and presents our new composite key management approach. Our metrics of authentication is presented in Section 3. The detailed design of the composite key management frameworks is described in Section 4. Section 5 presents the evaluations of composite key management with comparisons to existing approaches. Finally, we conclude with a discussion on future work in Section 6.

2. Key Management in Ad Hoc Networks

Many key management frameworks for ad hoc networks use either a virtual CA or certificate chaining. However, it is important to note that none of these approaches have been shown to provide effective solutions in diverse environments. This limitation mainly comes from the fact that most approaches try to adapt solutions from wired environments without adequately addressing the specific challenges in ad hoc networks. To understand why previous approaches fail, we define two underlying principles for key management in ad hoc networks: *node participation* and *the use of a trusted third party*. While most previous approaches adhere to one of these principles, it is often at the expense of the other, resulting in non-universal solutions. After defining each principle in detail, we discuss existing approaches with an emphasis on how they achieve or fail to achieve these proposed principles. Finally, we present a novel key management paradigm called *Composite Key Management*, which supports these principles through careful integration of key management mechanisms.

2.1. Two Principles for Ad Hoc Key Management

The *node participation* principle states that a key management framework for ad hoc networks should rely on a large number of nodes for availability, but a smaller group of nodes for security. Given the physical vulnerability of mobile nodes in ad hoc networks, it is not effective to burden a single node with the responsibility of providing a security service like key management. A natural way to address this problem is to distribute the security service over multiple nodes. In general, this set can span from a single node to all nodes in the network. However, blind and equal distribution of security functionality over too many nodes leads to a vulnerable system. This observation leads to two important questions as to the participation of nodes in key management. First, *How many of the nodes should participate?* The participation of a higher fraction of nodes in the network, can improve availability and fault tolerance. However, without careful consideration, higher participation can also lead to higher vulnerability. This leads to the second question: *How should the nodes participate?* When a security service is divided across a large number of nodes with equal responsibilities, the availability of the service increases since there are more nodes that an end user can contact. However, this improved availability also helps adversaries locate and compromise these nodes and eventually compromise the security of the service. Therefore, a blind and equal distribution of functionality to multiple nodes can degrade the overall security. Instead, core functionalities of the security service should be distributed to a restricted set of secure nodes, providing strong security and an acceptable level of availability. The rest of the nodes share lower level functionality to improve the availability of the core nodes. Compromising any of these low level nodes should not compromise overall security, but only affect the availability of the core service.

The use of a trusted third party (TTP) principle states that a key management framework should use a TTP to improve the quality of authentication of the framework. Without a clear trust anchor in the network, authentications can only rely on casual trust relationships. Since there are no guarantees about the behavior of participating nodes, any authentication based on such casual relationships cannot be trusted for security-sensitive applications. A TTP provides a trust anchor that can be used as the basis for further trust relationships. Since every node trusts the TTP, authentication provided by the TTP is trusted with a high level of confidence. Essentially, without trustworthy authentication, no further security service can be built to guarantee a high level of assurance. Therefore, using a TTP is crucial for any ad hoc network with strong security requirements.

2.2. Key Management Frameworks

Given these two principles for key management in ad hoc networks, we now discuss current approaches and how they succeed or fail in supporting these principles. We conclude this section with the presentation of *Composite Key Management* and discuss how it builds on existing approaches using these principles.

2.2.1. Virtual CA Approaches To address the unique challenges in ad hoc networks, several virtual CA approaches employ threshold cryptography to securely distribute the CA's functionality over multiple nodes [8, 12, 13, 15]. CA functionality is distributed in such a manner that an adversary must compromise a certain fraction of the key shares to compromise the virtual CA itself. At the same time, an end user need only access a subset of the distributed CA nodes to get certification services. Wu et al. first suggest a distributed CA based on threshold cryptography [12] and Zhou et al. propose its application to ad hoc networks [15]. Kong et al. [8] and Yi et al. [13] follow through by designing full key management frameworks. In both approaches, some subset of nodes are chosen to participate as members of the virtual CA. It is clear that all virtual CA approaches employ the use of a TTP principle.

While all virtual CA approaches appear to similarly address the node participation principle, the approaches differ in *how* they choose nodes to participate. Kong et al. proposed a virtual CA solution where every mobile node in an ad hoc network acts as a CA node and shares the responsibility of a CA [8]. This approach maximizes node participation by utilizing all nodes in the network, achieving very high availability. However, their solution is vulnerable to adversaries that can compromise a relatively small number of mobile nodes, and also to Sybil attacks [5]. Essentially, this approach violates the security component of the node participation principle by involving all nodes in the core security function. Yi et al. proposed MOCA [13], a generalized key management framework for all possible configurations of virtual CA approaches based on threshold cryptography. They suggest that the fraction of CA nodes should be kept to a relatively small to maintain strong security. This fits well with the security component of the node participation principle that limits the main key management functions to a small fraction of nodes. However, the MOCA framework sacrifices the first part of the principle, and so availability, by not involving the rest of the nodes in any part of key management. Essentially, Kong et al.'s approach sacrifices the security to achieve ubiquitous availability while MOCA sacrifices availability to maintain strong security.

2.2.2. Certificate Chaining Authentication by a chain of authorities has commonly been used in large scale dynamic networks without a single authority [7, 16]. In general, authentication is represented as a set of digital certificates that form a chain. Certificate chaining does not require heavy infrastructure or complex bootstrapping procedures and every node has identical roles and responsibilities. These characteristics of certificate chaining make it a potential candidate for key management in ad hoc networks, as realized by Hubaux et al. [4]. Certificate chaining achieves the maximum level of node participation, since every node can participates by issuing certificates to each other to populate the certification graph. However, every node shares the same responsibilities, limiting the security of the system. Additionally, certificate chaining fails to use any TTP. This lack of adherence to the principles leads to two main limitations.

First, since participating nodes operate in a best-effort manner, there can be situations when authentication cannot be provided. Essentially, a certification graph may not be populated enough to provide certificate chains for the given pair of nodes. Since there is no means to force mobile nodes to issue certificates and keep the certification graph dense enough, it is difficult to predict if any given authentication request can be fulfilled. As shown in two studies [3, 10] of PGP [16], a 1998 snapshot of the PGP certification graph that included 57582 nodes only had 3100 nodes (5%) in its largest strongly connected component (SCC) [10], while in a more recent snapshot, there is an even larger gap between the total number of nodes and the size of the largest SCC (2.5%) [3]. Essentially, PGP has one large SCC that contains a very small fraction of the nodes and the most of nodes are scattered to form a sparse graph. This gap is important since only members of the same SCC can authenticate each other.

Second, without relying on a TTP, any trust relationships must rely on the goodwill and the correct behavior of all participants. Any single misbehaving or malicious node participating in a certificate chain can taint the whole chain and invalidate the authentication. However, since there is no clear way to tell if a certificate chain includes any misbehaving nodes, the overall confidence value of certificate chains must be relatively low. To combat this problem, several enhancements have been proposed, including limiting the chain length and using multiple node-disjoint chains [11]. Despite these improvements, the level of assurance provided by certificate chaining may still not be strong enough to support strong security requirements.

2.3. Composite Key Management

It is apparent that an effective key management framework for ad hoc networks must include a secure TTP but still encourage participation from as many nodes as possible. To address both of these principles, we propose a novel paradigm for ad hoc key management called *Composite Key Management*, which uses a virtual CA and certificate chaining simultaneously in a single ad hoc network. The virtual CA in composite key management follows the suggestion by Yi et al. and uses only a small subset of more trustworthy and secure nodes for the virtual CA. With this design, composite key management can provide a TTP with strong security, satisfying the use of a TTP principle and the security component of the node participation principle. At the same time, the rest of the nodes participate in certificate chaining along with the distributed CA nodes to satisfy the availability component of the node participation principle, improving the availability and the coverage of the virtual CA to a level of ubiquitous presence. This combination of mechanisms can also improve the quality of authentication over pure certificate chaining since a certificate chain-based authentication can now rely on the TTP as a trust anchor, making the authentication inherently more trustworthy.

However, it is not simple to combine two heterogeneous approaches into a unified framework. Essentially, the meaning of an authentication result becomes more complex since end users must understand two different types of mechanisms and reason about interactions between them. To solve this problem, we propose a concise set of authentication metrics that encompass both virtual CAs and certificate chaining as well as the interactions between them. With this metric, an end user can easily calculate a trust value for a given authentication request to render decisions about whether or not to authenticate another node. In Section 3, we describe our proposed metrics and provide more details on composite key management in Section 4.

3. Metrics of Authentication

A metric of authentication is a tool used to calculate how much confidence or assurance can be put on an instance of authentication. In this section, we present a unified metric of authentication that can be applied to all major key management frameworks: PKI, certificate chaining, and our composite key management. Our metric is based on previous research in authentication metrics for certificate chaining [1, 2, 7, 9, 11, 16] and enhanced with support for a TTP and the composition of certificate chaining with a TTP.

A good metric should reflect several factors including the design of the underlying key management framework and the trust relationships among the entities.Metrics of authentication should be easy to reason about and simple to apply, preferably without human intervention. In many cases, authentication in an ad hoc network must rely on partial information about trust relationships. Therefore, a good metric must also be able to render a meaningful answer even with only partial information.

The main contribution of our metric includes (1) the extension of previous metrics to accommodate the TTP in the system, and (2) the capability to encompass composite key management frameworks, and (3) the simplicity to support easy and efficient application by end users. In the remainder of this section, we present our trust model, which captures trust relationships in the system as a graph, and then introduce our metric.

3.1. Trust Relationship Model

Entities in a distributed system express trust relationships by issuing certificates to each other. For example, if Alice believes that Bob holds key pair K1, Alice issues and signs a certificate containing Bob's identity and public key K1. This certificate can be presented to another user, who can then verify the signature on the certificate using Alice's public key and be assured that Alice vouched for the binding between Bob's identity and K1.

Our model captures such relationships in a *certification graph*, where a node represents a public/private key pair and an edge represents a digital certificate. An edge is coupled with two attributes: the *identity* of the key holder and the *confidence value*. The confidence value of a certificate expresses the level of confidence the certificate issuer has in

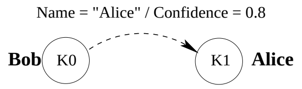

Figure 1. A Simple Certification Graph

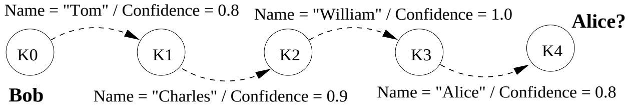
Name = "Tom" / Confidence = 0.8
Name = "William" / Confidence = 1.0
Alice?
K0
K1
K2
K3
K4
Bob
Name = "Charles" / Confidence = 0.9
Name = "Alice" / Confidence = 0.8

Figure 2. An Example Certificate Chain

the binding between the target user's identity and key. In our model, the confidence value can be between 0.0 (no trust) and 1.0 (absolute trust). For example, Figure 1 shows a simple certification graph with two nodes and one certificate. The edge from K0 to K1 denotes that the holder of the key K0 certifies Alice to be the holder of key K1 with 80% confidence. This simple model can represent any pure certificate chaining key management approach.

Since each edge in a certificate graph represents a simple trust relationship, multiple edges forming a *certificate chain* represent a complex relationship, defined by the attributes along each edge. Essentially, a certificate chain is a path of certificates from the user's node to the target node being authenticated. For example, Figure 2 shows a four hop chain from Bob to Alice.

3.2. Confidence Evaluation

If multiple certificate chains exist between a user and the target node, the user must select a chain or set of chains for authentication. We are currently investigating several options. But, in this paper, we only focus on the chain with the highest confidence value due to space limitations.

To determine the confidence value of a certificate chain, it is necessary to evaluate the attributes along every edge. For a given chain, the confidence values of all edges are multiplied to generate a *raw confidence value*, which is an intuitive measure of transitive trust. For example, if Alice trusts Bob with a confidence value of α and Bob trusts Charlie with a confidence value of β, the transitive trust that Alice has for Charlie is $\alpha * \beta$. Since our metric restricts confidence values to be between 0.0 and 1.0, multiplying confidence values never increases the overall value. Additionally, it is obvious that a long certificate chain has a higher chance to include a misbehaving node and thus the use of a long chain should be discouraged if a shorter alternate is available. Assuming each node in the network is equally likely to be malicious or compromised with a probability *p*, the probability that a chain of length *d* is intact can be denoted as $(1 - p)^{(d-1)}$ which we term the *attenuation factor* (not $(1-p)^d$ since the first hop is always from a trusted node). To accommodate this observation into our metric, the raw confidence value calculated from the previous step is multiplied by the attenuation factor, which decreases exponentially as the chain length grows, effectively discouraging the use of long chains. The raw confidence value of the chain in Figure 2 is $0.8*0.9*1.0*0.8 = 0.576$, while the attenuation factor with $p = 0.1$ is $(1-0.1)^{(4-1)} = 0.729$. Therefore, the final confidence value for the chain is $0.576 * 0.729 = 0.420$.

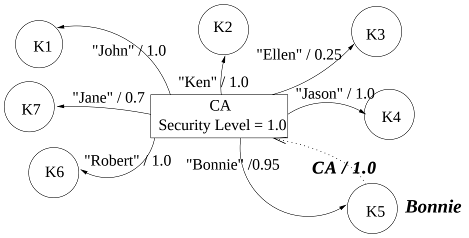

Figure 3. Certification Graph of a Pure CA-based System

Once the confidence value of the chain is calculated, the user can decide whether or not to grant the authentication request. The authenticating user can choose the threshold value for successful authentication on a per authentication basis, enjoying the full freedom to enforce their own security requirements.

3.3. Incorporating a TTP

To accommodate a TTP into a certification graph, we introduce a special *CA* node. For virtual CA approaches, the CA node represents all nodes that comprise the virtual CA. Since the CA is the trust anchor for the whole network, the security of a CA-based system solely depends on the security of the CA. Therefore, users should be able to express their own perception of the security of the CA. In our metric, the user's perception of the CA is expressed as the *security level* of the CA node in the user's local certification graph. Intuitively, there is an implicit edge from each user to the CA node. However, the confidence value of each edge can be different based on each user's perception of the security of the CA. Figure 3 shows an example local certification graph of a pure CA-based system from the view of a user. Assigning an appropriate security level to the CA node is a challenging problem. We are currently investigating various combinatorial measures based on the configuration of the virtual CAs and plan to incorporate these measures into our design in the future.

In a certification graph without a CA node, the authenticating user must be the first node in the chain. If the chain does not start at the user, the user has no trust relationship with the chain and cannot trust the chain. If the key man-

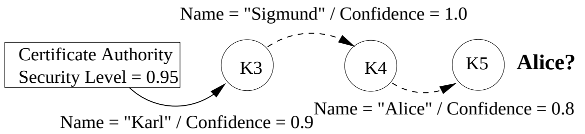
Name = "Sigmund" / Confidence = 1.0
Certificate Authority
Security Level = 0.95
K3
K4
K5
Alice?
Name = "Karl" / Confidence = 0.9
Name = "Alice" / Confidence = 0.8

Figure 4. A Certificate Chain Beginning at the CA

agement framework supports a CA, the CA is also considered as a trusted node and so can be the first node in a chain. However, if the chain includes a CA node, the chain's confidence value is multiplied by the security level of the CA node. In Figure 4, a certificate chain begins at a CA node with a 0.95 security level. The confidence value for this chain is 0.5832 after applying the attenuation factor, which is then multiplied by the CA's security level, yielding the final confidence value of $0.583 * 0.95 = 0.554$ for the whole chain.

4. Design of Composite Key Management

The composition of a virtual CA and certificate chaining requires deploying the two frameworks simultaneously and equipping the end users with the proper tools to use the composite framework. The metrics of authentication presented in Section 3 allow end users to understand potentially complicated authentication information provided by a composite key management framework. We can view the metric as the "glue" that binds all the components together. Since a virtual CA and certificate chaining are both self-contained approaches, they can be deployed in any manner possible: simultaneous deployment of both, adding a virtual CA to an existing certificate chaining system, or adding certificate chaining to an existing virtual CA framework. To better understand the interactions among the nodes in a composite framework, we first describe the three different types of nodes and clearly define their roles. We then list some specific examples of composite key management.

4.1. Node Types in a Composite Key Management

In composite key management, there are three types of nodes : CA nodes, nodes participating in certificate chaining, and client nodes that use the key management service. A single node can belong to more than one group.

- *CA Node*: A CA node carries a share of the virtual CA's private key and serves as one of the multiple nodes that comprise the virtual CA. A CA node is equipped with the capability to generate partial signatures using its key share, participate in certificate revocation and maintain a list of certificates issued by the virtual CA. For a detailed example of this type of node, we refer readers to previous works on distributed PKI in ad hoc networks [8, 13, 15].
- *Participant in Certificate Chaining*: A node participating in certificate chaining must be able to authenticate its neighbors, create and issue certificates for neighbors, and maintain the set of certificates it has issued. For a detailed example of this type of node, we refer readers to Hubaux et al. [4].
- *A Client*: Any client that makes authentication decisions must be able to understand certificates from both the virtual CA and from certificate chaining. Therefore, all client nodes must be equipped with the metric of authentication presented in the previous section. All authentication information is mapped to a local certification graph, which is used, along with the metric of authentication, by the client to calculate a confidence value for an authentication instance and decide on the authentication of the target node. This type of decision process allows individual nodes to apply their own criteria as to whether or not to authenticate on a per authentication basis.

4.2. Composition Examples

Since composite key management currently utilizes two types of techniques, it is useful to separate the effects of each technique on the other and study them in isolation. Therefore, we present example compositions based on each technique. By gradually adding in the other technique, we can observe the effects separately. Since the composition examples use a virtual CA and certificate chaining, there are two base certification graphs that need to be composed. Figure 5 represents the certification graph for the virtual CA component. All edges begin at the CA node and end at the end user nodes. Additionally, all edges are solid, indicating that these edges represent CA-issued certificates. Figure 7 represents the certification graph for the certificate chaining component. All edges are dashed arrows representing certificates are issued by peer nodes. These distinctions between the two types of edges are only for illustrative purposes and edges are not distinguished in the actual application of the metric.

The first composition uses certificate chaining to enhance the coverage of a virtual CA. The configuration of the certificate chaining component determines the limit on chain lengths. With 1-hop chaining, only nodes that have been certified by the virtual CA are allowed to issue certificates to other nodes. In this configuration, if a node wishes to acquire a certificate but cannot reach the virtual CA, the node can search its neighborhood to find any node that has been certified by the virtual CA. The original virtual CA certification graph in Figure 5 is augmented with several

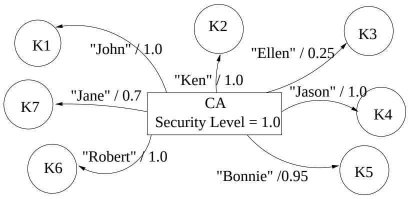

Figure 5. Standalone Virtual CA

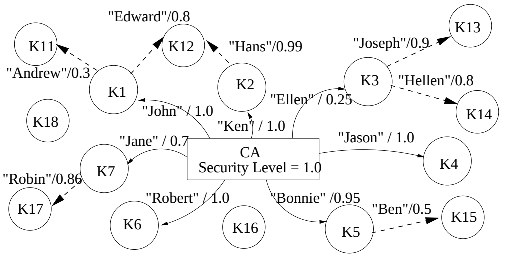

Figure 6. Virtual CA composed with 1-hop Certificate Chaining

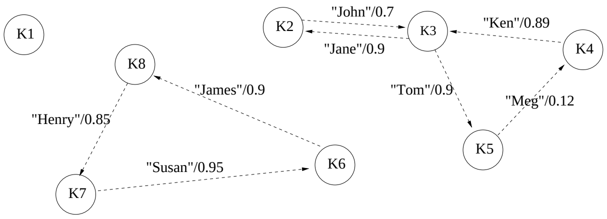

Figure 7. Pure Certificate Chaining

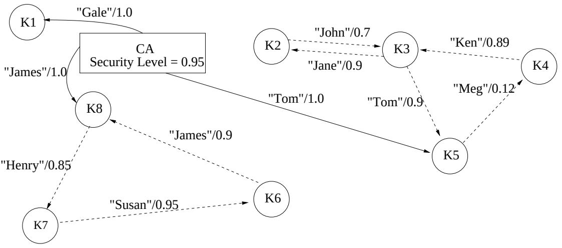

Figure 8. Certificate Chaining with CA-certified Nodes

1-hop chains to form the graph in Figure 6. Nodes with incoming dashed edges, like K11 and K12, are certified by CA-certified nodes, but not by the CA. In the original virtual CA certification graph, the average confidence value for all nodes is 0.843. In the composed graph, while the extended coverage of the composite framework covers six more nodes (K11, K12, K13, K14, K15, K18), the average confidence value decreases to 0.657 (with *p = 0.1* for attenuation) due to the lower confidence values of the newly added certificate chains.

The second composition begins with a pure certificate chaining component. A TTP is introduced by allowing CA-certified nodes to participate in certificate chaining. By design, a node certified by a CA is more trusted and can be used to create new chains with higher levels of assurance. The certification graph of the pure certificate chaining component in Figure 7 can be augmented with certifications from a virtual CA as shown in Figure 8. In the original certification graph in Figure 7, there are three SCCs and nodes can authenticate each other only within an SCC. For example, K7 cannot authenticate K3 because there is no certificate chain from K7 to K3. However, in the composed certification graph in Figure 8, K5 and K8 are certified by the CA and therefore trusted. K7 in the composed system can authenticate K3 by following a chain from (K5→K4→K3). The confidence values of certificate chains also increase due to the virtual CA. For example, the confidence value that K2 has for K5 is $0.7 * 0.9 * (1 - 0.1)^1 = 0.567$ (with *p = 0.1* for attenuation) in the original certification graph using the chain (K2→K3→K5). In the composed graph, the authentication has an increased confidence value of 0.95 following a direct chain from the CA (CA→K5). Such composition is simple and cost-effective and can enhance any certificate chaining system. We are currently studying the effect of this configuration on real-world certificate chaining systems like PGP [16].

5. Evaluation

We demonstrate the effectiveness of composite key management through two sets of experiments. We simulate stressful but realistic scenarios for a virtual CA or for certificate chaining and the effect of introducing composite key management.

5.1. Composing a Virtual CA with Certificate Chaining

By composing the virtual CA with certificate chaining, composite key management increases the availability and maintains strong security of the virtual CA. While composite key management can be applied to any kind of virtual CA scheme, we choose the MOCA virtual CA by Yi et al. [13] for this experiment. While MOCA adheres to the security component of the node participation principle, it has been shown that MOCA cannot achieve a 100% success ratio under stressful situations due to mobility and intermittent connectivity. In this experiment, 1-hop certificate chaining is used to augment the MOCA framework. Any node that has

Number of Total Nodes	150
Number of MOCA nodes	30
Crypto Threshold *k*	1, 2, 5, 10, 15, 25, 30
Network Area	1000m x 1000m
Simulations Time	600 seconds
Certificate Request Pattern	10 requests from 100 client nodes (Total 1000 requests)
Mobility	Max speed of 20m/s, 10 sec pause time

Table 1. Simulation Parameters for ns-2

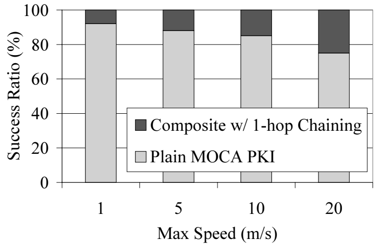

Figure 9. Success Ratio vs. Mobility, *k = 15*

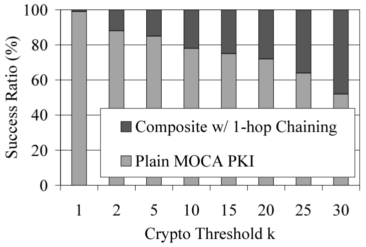

Figure 10. Success Ratio vs. Crypto Threshold k, max speed = 20 m/s

Max Speed (m/s)	1	5	10	20
MOCA Packet Overhead (pkts)	9234	130423	170375	190241
Chaining Packet Overhead (pkts)	189	256	332	503
Overhead Increase (%)	2	0.1	0.3	0.2

Table 2. Communication Overhead for Composite Approach, k=15

been certified by the MOCA framework can issue certificates to other nodes.

In our simulation set-up, out of 150 total nodes, 30 nodes are selected to serve as the MOCA nodes. To stress the MOCA virtual CA, we conducted two different types of simulations. We first evaluate the effect of mobility on the availability. Second, we evaluate the effect of the crypto threshold *k* by fixing the number of MOCA nodes in the network and increasing *k*. Crypto threshold is a common parameter to any distributed security service relying on a quorum of nodes to reach a decision. In this case, *k* is the minimum number of MOCA nodes a client must contact to receive certification service. In all simulations, when a node requests a certification service, the node first tries to contact the virtual CA. If that fails, the node probes its 1-hop neighborhood to check if there are any CA-certified nodes. All simulation results are an average of five different scenarios with the same parameters in different topologies. Simulation parameters are shown in Table 1.

To evaluate the effect of composing MOCA with 1-hop certificate chaining on the success ratio, we fixed the crypto threshold *k* to 15 and gradually increased mobility. As shown in Figure 9, the success ratio of MOCA degrades from 92% to 78% as mobility increases. However, the 1-hop certificate chaining always succeeds in filling the gap and improving the success ratio to stay between 99.2% and 100%. Similarly, Figure 10 shows results from varying the crypto threshold *k* with a fixed maximum speed of 20m/s. When $k = 1$, a client only needs to contact one MOCA node and the success ratio is 99%. However, as *k* increases, a client node must contact more MOCA nodes and the success ratio decreases. In the extreme case of $k = 30$, the success ratio drops to 52%. When 1-hop chaining is added, a 99-100% success ratio is achieved. As shown from these two evaluations, even the simplest form of certificate chaining can alleviate the availability problems of a virtual CA.

The benefits of 1-hop certificate chaining comes with negligible overhead. Both communication and computation overhead have been observed to be negligible since chaining is only invoked when MOCA authentication has failed. Communication overhead is localized to 1-hop neighbors and each certificate request consists of a single broadcast request packet and one or more reply packets. The packet overhead from a simulation with fixed $k = 15$ and varying mobility is shown in Table 2. Results from varying simulation parameters show similar trends. Packet overhead stays under 2% of the MOCA overhead in all cases.

This composite approach satisfies both principles without sacrificing security or availability. This demonstrates that limiting the core security functionality of the CA to a small fraction of trusted nodes can maintain high security and availability at the same time by using the remaining nodes to provide extended coverage of the virtual CA.

5.2. Composing Certificate Chaining with a TTP

The fundamental problems with certificate chaining stem from the fact that the certification graph is generated by the voluntary actions of individual nodes. We evaluated the effect of composing certificate chaining with a TTP using several realistic scenarios that stress this limitation.

Since the composite approach is aimed at ad hoc environments, we generated certification graphs using a popular ad hoc mobility pattern generator `set-dest` with corrections for the speed-decay problem [14]. All mobility patterns include 100 nodes moving in 5000m by 5000m area for 600 seconds in simulation time. When any two nodes stay in each other's transmission range for longer than one minute, we assume that these two nodes always issue certificates to each other. A one minute threshold is chosen to give nodes enough time to check each other's identity as well as to create and issue certificates. Such certification graphs are as dense as possible for the given mobility patterns. To simplify the evaluation, every certificate is issued with a maximum confidence value of 1.0. These two choices allow pure chaining to achieve the best possible performance, setting the baseline as high as possible for fair demonstration of improvements from composition. The injection of CA-certified nodes into the certification graph is achieved through random sampling. Nodes are randomly labeled as CA-certified up to the target fraction of CA-certified nodes. Since the simulation area is relatively large compared to the number of deployed nodes, the density is not very high, resulting in sparse certification graphs. We study the effect of varying the fraction of certified nodes and the maximum speed. Pause time in all patterns is fixed to 60 seconds.

Any variation of certificate chaining can be used for composition. However, the results from our experiment represent the best achievable results for any certificate chaining approach since all nodes are provided with complete knowledge of the full certification graph. For example, an approach like Capkun et al. [4] divides up the certification graph across multiple nodes. It is highly likely that a subset of mobile nodes can only recreate a part of the certification graph. In such cases, the resulting certification graph will be sparser than the full graph and the performance of pure chaining will degrade.

To evaluate composing certificate chaining with a TTP, we consider two metrics: the number of successful authentications and the quality of authentication. For the quality metric, we measure the average confidence value from all chains resulting in successful authentications.

In the first set of experiments, the fraction of certified nodes is increased from 0% to 100% with the maximum speed fixed at 10m/s. With no certified nodes in the network, only 44% of all possible pairs of nodes can authenticate each other in pure certificate chaining (See Figure 11). As the fraction of certified nodes increases, the number of successful authentications for the composite approach increases significantly and reaches a 100% success ratio when every node in the network is certified. With 10% of the nodes certified, the composite approach provides 11% more authentications, while with all nodes certified, the improvement goes up to 128%.

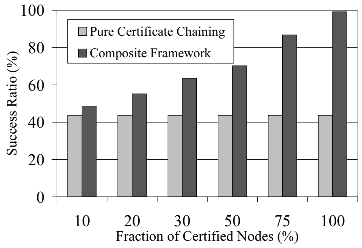

Figure 11. Success Ratio vs. Fraction of Certified Nodes

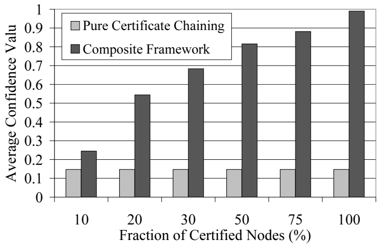

Figure 12. Average Confidence Value vs. Fraction of Certified Nodes

Figure 12 presents the average confidence values from all certificate chains used for successful authentications. As clearly displayed in Figure 12, the average confidence value in the composite framework is 66% to 570% higher than in pure certificate chaining. This is due to fact that with many certified nodes in the network, the length of the certificate chains decreases, resulting in higher confidence values.

Figures 13 and 14 present similar results for varying mobility. As the maximum speed increases from 0 m/s to 25 m/s, the success ratio of pure certificate chaining improves from 0.8% to 94%. As nodes move faster, they travel farther and have a higher chance to meet more nodes, resulting in a denser certification graph. The success ratio of the composite approach also increases for the same reason. However, in this experiment, Figure 14 is more interesting. While the average confidence value of pure chaining stays within bounds, it gets worse in the composite approach. Originally, the confidence value from the composite approach is high since most authentications rely on CA-certified nodes. However, a denser certification graph through high mobility improves the overall success ratio by creating more chains with lower confidence values and these new chains reduce the the average confidence value.

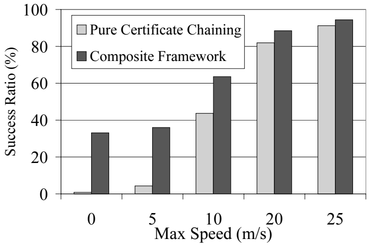

Figure 13. Success Ratio vs. Mobility, with 30% of certified nodes

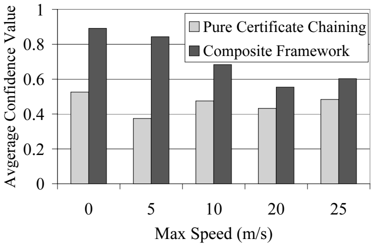

Figure 14. Average Confidence Value vs. Mobility, with 30% of certified nodes

6. Conclusions and Future Work

In this paper, we propose a novel approach for key management in ad hoc network called *Composite Key Management*. Based on the observation of the limitations of existing solutions, we first present two key principles for ad hoc key management: *node participation* and the *use of a trusted third party*. Based on these two principles, detailed mechanisms to implement a composite key management framework are presented with a set of new metrics of authentication that glues each component. Using two representative configurations of composite key management, we demonstrate the effectiveness of composition. Through extensive simulation studies, we demonstrate that composite key management satisfies both principles for ad hoc key management and can provide flexible, modular, and adaptive key management services for ad hoc networks.

Composite key management still leaves many interesting questions to be studied. First, we plan to find a clear and intuitive way to calculate the security level of a virtual CA, which must take into account several factors including the security of individual distributed CA nodes, the spatial distribution of the distributed CA nodes, and the state of the network. Second, using only the highest confidence certificate chain in our current metric does not fully exploit the information contained in a certification graph. We are investigating alternative approaches including the use of multiple node-disjoint chains. Third, the certificate chaining composed with a virtual CA can be also realized in wired networks. We plan to investigate the effect of using certified nodes and the certification graph from the PGP system. Finally, we plan to extend the metrics of authentication presented in this paper and study the mathematical and combinatorial characteristics of the proposed metrics using graph theory.

References

[1] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proceedings of the Conference on Computer Security*, 1994.

[2] A. D. Birrel, B. W. Lampson, R. M. Needham, and M. D. Scheroeder. A global authentication service without global trust. In *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, 1986.

[3] S. Capkun, L. Buttyán, and J.-P. Hubaux. Small worlds in security systems: an analysis of the pgp certificate graph. In *Proceedings of the 2002 workshop on New security paradigms*, pages 28–35. ACM Press, 2002.

[4] S. Capkun, L. Buttyán, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1), 2003.

[5] J. Douceur. The sybil attack. In *Proceedings of IPTPS 02 Workshop*, Mar. 2002.

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

[7] S. T. Kent. Internet privacy enhanced email. *Communications of ACM*, (8):48–60, Aug. 1993.

[8] 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.

[9] R. Levien and A. Aiken. Attack-resistant trust metrics for public key certification. In *Proceedings of the 7th USENIX Security Symposium*, Jan. 1998.

[10] N. McBurnett. PGP web of trust statistics. URL: http://bcn.boulder.co.us/ñeal/pgpstat/, 1998.

[11] M. K. Reiter and S. G. Stubblebine. Authentication metric analyis and design. *ACM Transactions on Information and System Security*, 1999.

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

[13] 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)*, Apr. 2003.

[14] 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.

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

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