Composite Trust-based Public Key Management in Mobile Ad Hoc Networks

Jin-Hee Cho
U.S. Army Research Laboratory
2800 Powder Mill Rd.
Adelphi, MD 20783
jinhee.cho@us.army.mil

Kevin S. Chan
U.S. Army Research Laboratory
2800 Powder Mill Rd.
Adelphi, MD 20783
kevin.s.chan@us.army.mil

Ing-Ray Chen
Virginia Tech
7054 Haycock Rd.
Falls Church, VA 22043
irchen@vt.edu

ABSTRACT

Public key management in mobile ad hoc networks (MANETs) has been studied for several decades. Yet no single solution has completely resolved well known design challenges resulting from the unique characteristics of MANETs. These challenges include no centralized trusted entities, resource constraints, and high security vulnerabilities. This work proposes a fully distributed trust-based public key management approach for MANETs using a soft security mechanism based on the concept of trust. Instead of using hard security approaches, as in traditional security techniques, to eliminate security vulnerabilities, our work aims to maximize performance by trading off risk (i.e., security vulnerability) for trust. In this work, we propose a composite trust-based public key management (CTPKM) with no centralized trust entity with the goal of maximizing performance (e.g., service availability or efficiency) while mitigating security vulnerability. Each node employs a trust threshold to determine whether or not to trust another node. Each node's decision making using the given trust threshold affects performance and security of CTPKM. Our simulation experimental results show that there exists an optimal trust threshold that can best balance and meet the conflicting goals between performance and security, exploiting the inherent tradeoff between trust and risk.

General Terms

Algorithms, Management, Measurement, Performance, Design, Experimentation, Security.

Keywords

Public key management, mobile ad hoc networks, trust, trustworthiness, private key, public key, certificate, certificate authority, risk.

1. INTRODUCTION

In resource-constrained network environments such as mobile ad hoc networks (MANETs), it is not feasible or efficient to employ cryptographic techniques for key management due to high computation and communication overhead as well as network dynamics that could require frequent key reassignments. In addition, the unique nature of MANETs does not allow any centralized trusted certificate authority (CA) to deal with all key management operations including key generation, distribution, update, and revocation. Essentially, we are not able to build a perfect system even using hard security approaches (e.g., encryption or authentication techniques) to meet the dual goals of efficiency and security due to their inherent tradeoff. In this work, we take a soft security approach by applying the concept of trust in order to meet security requirements while maximizing performance.

Specifically, we propose a composite trust based public key management (CTPKM) for MANETs. The proposed protocol is designed to meet a required level of security (e.g., exposed risk) as well as to meet performance requirements (e.g., service availability or communication overhead), without relying on trusted third parties such as CAs. The proposed protocol aims to be: (1) resilient against misbehaving nodes (i.e., untrustworthy nodes) in the network in order to maintain minimum security vulnerability (i.e., a small number of nodes using an undetected compromised key); (2) available in service provision in the presence of threats (i.e., a sufficient number of valid/correct public keys kept in each node); and (3) efficient in minimizing communication overhead incurred by the proposed key management operations.

The main contributions of this work are as follows. First, different from hard security that strictly adheres to credential-based security policies regardless of network dynamics or security requirements, this work uses a soft security approach based on the concept of trust to exploit the tradeoff between risk (i.e., security vulnerability) and trust (i.e., performance). We use the definition of trust as the willingness to take a risk to achieve a given task [16]. However, at the same time, it is critical to minimize risk in order to achieve the task successfully. This work examines a way to meet an acceptable risk (not to eliminate risk but to be tolerable against security vulnerability or risk) while maximizing performance including high service availability (i.e., more valid public keys) and low communication overhead. Second, we use a composite trust metric considering various aspects of trust in an entity, instead of considering a single dimension of trust; this composite nature of trust has not been addressed particularly for MANET environments in the literature. Third, the proposed public key management scheme is a fully distributed algorithm that can best fit a MANET environment without requiring any centralized trusted CA. Lastly, by exploiting the inherent tradeoff between trust and risk, we identify an optimal trust threshold (for differentiating trustworthy vs. untrustworthy in our protocol) under which the key management service availability is maximized.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the system model. To be specific, we discuss the trust model in Section 3.1, the proposed public key management scheme (CTPKM) in Section 3.2, the attack model in Section 3.3, and the performance metrics in Section 3.4. We show our numerical results and analysis in Section 4. We conclude our paper and outline future work directions in Section 5.

2. RELATED WORK

In the literature, public key management for MANETs has been explored mainly with the following schemes: certificate-based, identity-based (ID-based), threshold cryptography, certificate-less cryptography, and hybrid mechanisms combining multiple existing methods.

Certificate-based public key management approaches require public keys to be distributed where the receiving party should be able to authenticate the received key based on the certificate of the public keys. Thus, a trusted CA is required to deal with key management operations including key generation, distribution, revocation, and update [8]. For MANETs without trusted CAs, certificate-based approaches should operate in a self-organized way. Capkun et al. [3] proposed a certificate-based public key management where each node issues pairs of its own public and private keys and the certificate of the public key with a limited validity period so as to deal with network partitions in MANETs. However, finding a valid certificate chain generates high communication overhead. In addition, explicit key revocation requires high communication overhead while implicit key revocation leads to security vulnerability until the private key compromise is detected. Chang and Kuo [4] proposed a two-step secure authentication protocol for multicast MANETs. In order to deal with key management, they used the highest trustworthy node as a CA and the second highest trustworthy node as a backup CA. However, different from our work, they considered trust is already in place and static over time.

Huang and Wu [10] proposed a certificate path discovery algorithm for MANETs based on the hierarchical PKI structure using multiple CAs with no specific trust framework given. Huang and Nicol [11] proved that the shortest certificate chain does not guarantee the most trustworthy path to obtain the public key of a target node due to different trustworthiness observed in each intermediate node on the certificate chain. Vinh et al. [18] employed a group header for public key management in a group communication system where the group header is selected based on trust.

Shamir [15] proposed the concept of *ID-based public key management* scheme which generates a public key based on the ID of the node (e.g., IP or email address) and its corresponding private key generated by a trusted CA. The weakness of the ID-based scheme is its known high security vulnerability under the compromise of the trusted CA. Even with this disadvantage, the ID-based public key management is popularly applied in resource-restricted network environments since it reduces communication overhead by reducing the size of secret information (i.e., ID) to generate a public key.

In *threshold cryptography* [9], the private CA key is distributed over a set of server nodes through a (k, n) secret sharing scheme. The private CA key is shared between n nodes in such a way that at least k nodes must cooperate in order to reveal the key. However, a central trusted dealer exists to select servers as the coordinators for key management, resulting in a single point of failure. In addition, the inherent weakness of the secret sharing scheme is the substantial delay when the set of trustworthy server nodes are not found to generate the private CA key. Besides, when the CA is compromised, the system is compromised. Dahshan and Irvin [8] proposed a trust-based threshold cryptography scheme for MANETs in which a private key can be generated by a share of keys obtained from a set of trustworthy 1-hop neighbors. However, the weaknesses described above cannot be solved particularly in low node density or highly hostile network environments, while our protocol concerns resilience against node attack behaviors.

To cope with the communication overhead incurred in exchanging certificates, Sattam et al. [13] introduced the concept of *certificate-less public key cryptography* (CL-PKG). Compared to traditional public key cryptographic systems, CL-PKC does not require the use of certificates to ensure the authenticity of public keys. A trusted third party (TTP) having a private master key manages the authenticity of public keys.

Some researchers proposed *hybrid public key management* mechanisms that combine the features of multiple existing schemes. Sun et al. [17] combined ID-based key management and threshold cryptography to propose a key management solution for MANETs without any trusted CAs. Secret shares are distributed by each node and a trusted CA is not needed to generate a master private key. However, as is the nature of threshold cryptography, key updates require high communication overhead due to the communications of secret shares upon every key update. In addition, it is not clear how one should pick a set of nodes for private key generation. Li et al. [12] used Sattam's certificate-less public key cryptography to eliminate the key escrow problem and employed threshold cryptography to obviate the need for a centralized third party. However, this work still assumes that there exist multiple trusted Key Generating Centers (KGCs) for managing shares of a master private key. Moreover, the selection of KGCs is not based on their trustworthiness, but rather they are the ones with the smallest ID. Thus, the countermeasure for the compromise of KGCs is not considered.

3. SYSTEM MODEL

We consider a MANET with no centralized trusted entities (i.e., no centralized CA) to deal with public key management. Nodes are modeled with heterogeneous characteristics with vastly different speed (v_i), monitoring capability (which affects the false positive/negative probability of detection denoted by P_{fp}/P_{fn}), group join/leave rate (λ/μ), and a different trust level in integrity, competence, or social contact. For trust properties, we discuss them in detail in Section 3.1 below. We assume that a node moves around based on random mobility over the operational area with its given speed. A node's private key may also be compromised when the node possessing the private key itself is not compromised. We assume the private key compromise event arrives exponentially with the rate λ_{comp}. A node's trust is evaluated based on multiple aspects of trust: competence, integrity, and social contact. We describe each trust component in Section 3.1.

An entity (or node) is assumed to be a device carried by a human (e.g., dismounted soldiers or disaster rescue staff). Each entity is able to communicate with other entities using public/private key pairs obtained through the proposed CTPKM. A node will use a certain trust threshold to interact with another node. That is, given a trust threshold T_{th}, a node will assume a certain amount of risk and communicate with another node whose trust level is equal to or above T_{th}. We investigate the impact of trust threshold on performance metrics. Our proposed protocol satisfies the requirements of self-organized and distributed key management for MANETs as discussed in [7]:

- No single point of failure, i.e., no trusted third party is required;
- Resilience with low-security vulnerability in the presence of hostile entities, i.e., little exposure of a compromised key;
- High service availability, i.e., a sufficient number of valid public keys kept in each node; and

- Scalability in terms of low communication overhead for obtaining a valid public key.

3.1 Trust Model

We consider three trust components: competence, integrity, and social contact. We capture the competence and integrity characteristics from communication networks while deriving social contact characteristic from social networks. The three trust components are:

- **Competence (C)**: This refers to an entity's capability to serve received requests in terms of a node's cooperativeness and availability. Availability may be affected by network conditions such as link failure, energy depletion, and voluntary or involuntary disconnection (i.e., leaving the network). This is measured by the ratio of the number of positive experiences to the total experiences in packet forwarding.
- **Integrity (I)**: This is the honesty of an entity in terms of network attack behaviors such as fake identity dissemination (e.g., compromise another node's private key by identity or Sybil attack), false recommendation, message modification or forgery. This is computed as the number of positive experiences over the total experiences related to compliance of a protocol.
- **Social Contact (SC)**: This is defined as the number of nodes that a node encounters during a trust update interval T_{update} over the total number of nodes in the network. If an entity has high SC, it is more likely to disseminate information quickly to the network, compared to the ones with low SC. An entity's mobility pattern may affect trust in this component.

We assume that a node's trust profile is available, describing its inherent behavior patterns that can be scaled in [0, 1]. In our experiment setting, we generate the trust profile based on uniform distribution with the range in [GB, 1] where GB is the lower bound of good behavior. In this work, trust is being used for decision making, including obtaining a certificate of a public key, distributing a public key, requesting a public key of a target node, and providing a public key requested. The reasons we pick the above three trust components are (1) with competence trust, we assure fast propagation of public keys; (2) with integrity trust, we increase the probability that public keys propagated are valid/correct; and (3) with social contact trust, we increase the probability of finding a valid public key from nodes having many social contacts.

In this work, we build CTPKM on top of a trust model developed in [6] in which we examined the optimal trust chain length (TC) that maximizes trust accuracy (or minimizes trust bias) based on the difference (absolute value) between actual trust values (ground truth) and estimated trust values. Here we describe the key part of the trust model in [6]. Trust value is scaled between 0 and 1 as a real number. Trust of each trust component X is computed based on the aggregation of both direct and indirect evidences. Trust of node j (trustee: trusted party) evaluated by node i (trustor: trusting party) in trust component X is:

if ($|R_j| > 0$ && $HD(i,k) \leq TC$)

$$T_{i,j}^{X}(t) = \alpha T_{i,j}^{D-X}(t) + (1-\alpha) T_{i,j}^{ID-X}(t) \qquad (1)$$

else $T_{i,j}^{X}(t) = \gamma T_{i,j}^{X}(t-\Delta t)$

$T_{i,j}^{D-X}(t)$ is direct trust based on direct observations and $T_{i,j}^{ID-X}(t)$ is indirect trust based on recommendations from 1-hop neighbors of node j. R_j is the set of recommendations correctly received from node j's 1-hop neighbors. The availability of recommendations ($|R_j| > 0$) is affected by the receipt of the correct recommendations that are affected by packet dropping behaviors and integrity (e.g., dishonest behaviors) of a node. $HD(i,k)$ is the number of hop distance between nodes i and k where node i is a trustor (requestor) and node k is a recommender of node j (1-hop neighbor of node j). In order for the new indirect evidence to be used for trust update, recommender node k for node j should exist within the TC number of hops from node i and the correct recommendation should arrive safely in node i. We will use the optimal TC identified in this work, but do not investigate this in detail because this is already examined in [6]. The correctness of the recommendations is ensured by referring to direct opinions of referral recommenders (forwarding the original recommendation) attached to the original message with any detection error of the intermediate nodes forwarding the recommendation [6]. α and (1- α) are the weights for direct and indirect evidences respectively where α is between 0 and 1. We observe that when the weight (α) for direct evidence is low, high trust accuracy is observed, or vice-versa. This is because only correct recommendations based on unanimous agreement by all intermediate nodes that pass the recommendation are used as indirect evidence while new direct evidence cannot be collected easily due to node mobility. When no correct recommendations are received from recommender k located within TC hops from node i, trust decays with a decay factor γ over Δt, the periodic trust update interval T_{update}.

The direct trust of node i in node j on trust component X at time t, $T_{i,j}^{D-X}(t)$, is computed as:

$$T_{i,j}^{D-X}(t) = \begin{cases} T_{i,j}^{D-X}(t) \text{ if } HD(i,j) == 1 \\ \gamma T_{i,j}^{X}(t-\Delta t) \quad \text{otherwise} \end{cases} \qquad (2)$$

$HD(i,j)$ is the number of hop distances between nodes i and j. Thus, when nodes i and j are encountered as 1-hop neighbors during the time period $(t-\Delta t)$, node i can collect direct evidences based on its own observations or experiences. When nodes i and j are distant with more than 1 hop distances, node i relies on its past experience to assess the direct trust of node j.

The indirect trust of node j evaluated by node i on trust component X at time t, $T_{i,j}^{ID-X}(t)$, is obtained by:

$$T_{i,j}^{ID-X}(t) = \begin{cases} T_{k,j}^{X}(t) \quad \text{if } |R_j| > 0 \\ \gamma T_{i,j}^{X}(t-\Delta t) \quad \text{otherwise} \end{cases} \qquad (3)$$

When node i receives correct recommendations with $|R_j| > 0$, node i fully employs them to assess the indirect trust. If R_i is an empty set, node i will use its past experience $T_{i,j}^{X}(t-\Delta t)$ due to no correct recommendations received. For more details on the trust metric, refer to [6].

3.2 Composite Trust-based Public Key Management (CTPKM)

This work addresses the communications required to exchange confidential information between two nodes using public/private key pairs. Now we discuss the core operations of CTPKM.

3.2.1 Key Generation

For a self-organized public key management, each node generates its own public/private key pairs periodically. A node uses its private key to decrypt received messages while other nodes use the public key to send messages to the node. A node updates its public/private key pair when the expiration time is reached or the key pair should be updated due to the compromise of a current private key. Before distributing a key pair, a node should be able to obtain the certificate of the public key from a trustworthy node. The expiration time of a new key pair is randomly picked by each node based on exponential distribution with mean x. We discuss these in the following sections.

3.2.2 Public Key Certificate Issuance

Each node asks a 1-hop neighbor who has a trust value equal to or greater than the given trust threshold (T_{th}) for integrity trust (i.e., $T^{I}_{m1,i}(t) \geq T_{th}$), to be an issuer of the certificate of the public key; we call this node a neighborhood trustworthy certifier (NTC). The NTC should also decide whether to issue the certificate based on the trustworthiness of the requestor in integrity trust using T_{th}. That is, there should be a mutual trust relationship between a certificate requestor and an issuer in integrity trust. A node receiving a public key of another node will assess the authenticity of the received public key based on integrity trust of the NTC based on T_{th}. The requesting node is not able to obtain the certificate of its public key if its integrity trust level is below T_{th}. Recall that trust values are dynamically changing over time. The trust threshold T_{th} affects the decision making of a requesting node on whom to select as its NTC. If a low T_{th} is used, even a relatively untrustworthy node can issue and certify the public key to others. As a result, an attacker can disseminate many invalid/incorrect public keys so as to generate unnecessary communication, resulting in a waste of network resources.

3.2.3 Public Key Distribution

After a node obtains its public key certificate, the node disseminates the public key with the certificate to a set of its 1-hop neighbors whose trust values are equal to or greater than T_{th} for all three trust components. That is, a trustworthy 1-hop neighbor k of node i should meet the following conditions:

$$T^{C}_{i,k}(t) \geq T_{th}, T^{I}_{i,k}(t) \geq T_{th}, T^{SC}_{i,k}(t) \geq T_{th} \qquad (4)$$

where $T^{C}_{i,k}(t)$, $T^{I}_{i,k}(t)$, and $T^{SC}_{i,k}(t)$ are the subjective trust of node k evaluated by node i for competence, integrity, and social contact trust, respectively.

Each node i periodically disseminates its public key to the set of 1-hop neighbors (k's) where the members of the set may change as their trust values change over time. Since the nodes are mobile, if a node has a high mobility rate, it may have more chances to obtain public keys of other nodes, and vice-versa. Selecting the right set of neighboring nodes is critical in revealing less security vulnerability while obtaining valid/correct public keys. When a public key is distributed to an untrustworthy node, the untrustworthy node may not provide other nodes' correct public keys. It may even provide incorrect public keys of others, increasing communication overhead by disseminating incorrect public keys and leading to high security vulnerability where the untrustworthy node may leak confidential information to an attacker who has obtained the private key of the owner of the public key.

When node i disseminates its public key and the certificate to the set of 1-hop neighbors based on Equation 4 (called 'trustworthy 1-hop neighbors' hereafter), the packet consists of the following items:

$$\left[\left(C^{NTC\,(i)}_{K_{i,public}}\right)_{K_{i,private}}, K_{i,public}\right] \qquad (5)$$

$C^{NTC\,(i)}_{K_{i,public}}$ is the certificate of node i's public key signed by the NTC's digital signature including the information on the owner ID (node i), the NTC's ID, and expiration time for the valid period of the public key. Note the notation NTC (i) means a NTC who issues the certificate of node i's public key. $K_{i,private}$ is node i's private key, and $K_{i,public}$ is node i's public key. The certificate is encrypted by the private key of node i ($K_{i,private}$). The receiving node will be able to decrypt the message with the provided public key, $K_{i,public}$. Otherwise, the message will be discarded. If the receiving node can decrypt the message, it will check the trustworthiness of the NTC in integrity trust with T_{th} to ensure the authenticity of $K_{i,public}$. If the NTC is compromised, a fake public/private key pair to perform a fake identity or Sybil attack (with fake ID) may be generated. Integrity trust check for NTC minimizes the chance of this attack. Each node distributes its public key with the certificate to the selected set of trustworthy 1-hop neighbors based on Equation 4 periodically.

Some nodes may not have the public key of a particular node it wants to communicate with because it has not encountered the node as a 1-hop neighbor. In this case, a node can request the target node's public key to its trustworthy 1-hop neighbors based on Equation 4. If any of the trustworthy 1-hop neighbors (m_1) has the public key of the target node, then it will provide the public key to node i. Node m_1 decides whether or not to provide the public key of the target node based on the trust assessment of node i (requestor) in integrity trust (i.e., $T^{I}_{m1,i}(t) \geq T_{th}$). If node m_1 decides to provide the public key of the target node, the returning message includes the following items:

$$\left[C^{NTC\,(TN)}_{K_{TN,public}}, K_{TN,public}, ID_{m1}\right]_{K_{i,public}} \qquad (6)$$

Node m_1 encloses the public key of the target node (TN) ($K_{TN,public}$), the TN's public key certificate encrypted with the public key of node i ($K_{i,public}$), and its ID (ID_{m1}). All of this information is encrypted with $K_{i,public}$. When the requestor receives this message, it will save the public key of TN based on integrity trust of the provider m1. Notice that this returning message is encrypted by the requestor (node i)'s public key so only the requestor can decrypt this message.

If any of the trustworthy 1-hop neighbors (m_1) does not have the public key of TN, it will forward the request message to the set of its trustworthy 1-hop neighbors that meet the conditions in Equation 4. The delegated request message includes:

$$\left[C^{NTC\,(i)}_{K_{i,public}}, K_{i,public}, ID_{TN}\right]_{K_{m1,public}} \qquad (7)$$

$C^{NTC\,(i)}_{K_{i,public}}$ is defined in Equation 5. ID_{TN} is the ID of the TN, and $K_{m1,public}$ is the public key of node m1 who is a trustworthy 1-hop neighbor of the delegating node. The node receiving the delegated request message from m_1 decrypts the message with its private key, checks if it has the public key of the TN, and checks if the requestor passes the integrity test (i.e., $T^{I}_{m1,i}(t) \geq T_{th}$). If yes, then it sends the public key of TN to the original requestor (node i) by a returning message following the format specified in Equation 6.

If an intermediate node forwarding the request message is uncooperative or compromised, the request message can be dropped, therefore many nodes may not have valid public keys. A less trustworthy 1-hop neighbor may even provide more incorrect public keys. Therefore, the trust threshold (T_{th}) affects how many valid/correct public keys a node can use. We aim to identify an optimal T_{th} that generates a sufficient number of valid public keys that a node keeps while reducing communication overhead (not forwarding the public key requests to untrustworthy nodes) and security vulnerability (reducing the use of a public key whose private key is compromised). In CTPKM, when a trustworthy intermediate node that meets the conditions in Equation 4 has a valid public key of the target node, it can provide the public key of the target node to the requestor. This reduces communication overhead significantly in key distribution.

3.2.4 Key Revocation and Update

The private/public keys of a node will be revoked after the valid period expires. Since the certificate includes the information on expiration time, key revocation due to the passed valid period will be implicitly known to other nodes in the network. Before the valid period is past, a node's 1-hop neighbors can serve as verifiers and apply majority voting to detect if the node's private key is compromised. If the private key is deemed compromised (when the majority of the neighbors vote against it), the node, being as the owner of the private key, must notify the key compromise event to all nodes in the network. If the owner of the key itself is compromised and does not disseminate the key compromise message, its neighbors will decrease the trust value of the node to hinder the node from reissuing a new public/private key pair. In CTPKM, if a node does not maintain a certain level of trust, it cannot obtain the certificate of its public key. Therefore, there is a very low chance for an untrustworthy node to issue its public key with a valid certificate.

If an attacker compromises the private key of a node, this may introduce security vulnerability that other nodes may send confidential information using the public key to the owner of the compromised private key before the compromised private key is detected. A compromised private key can be detected when 1-hop neighbors of the attacker detect fake identity attacks and receive the public key of another node from the attacker. When the compromised private key is detected and this event is disseminated to all nodes in the network, the owner of the key must issue a new pair of public/private keys. In this work, we only consider the case that a node's private key is compromised by an attacker. When a node itself is compromised, CTPKM hinders its normal communication with other nodes by means of the compromised node's low trust value assessed by other nodes.

3.3 Attack Model

We consider the following attacks in MANETs: (1) *packet dropping* which interrupts service availability; (2) *message modification/forgery* which hinders effective communication; (3) *fake identity* (or impersonation attack) which breaks information confidentiality between two nodes; (4) *false recommendation* which promotes inaccurate trust evaluation; and (5) *compromise of private key* which allows an attacker to decrypt confidential information sent by other nodes using the associated public key. The compromise of private key can happen when a node impersonates another node to intercept all information forwarded to the original owner of the private key. We assume that impersonation attack can be detected by using mechanisms such as radio frequency fingerprinting or user mobility profiling [1]. We assume that each node is preinstalled with a detection mechanism to monitor misbehaviors of each node with known detection error probabilities (i.e. false positives or negatives). This will affect the accuracy of direct evidence for integrity trust (cases 2-5 above) and competence trust (case 1 above).

3.4 Metrics

We use the following metrics to assess performance of CTPKM:

- **Information Risk ($\mathbf{N_{CK}^{risk}}$)**: This indicates the number of attempts that untrustworthy nodes (due to low integrity or detection error) disseminate a message using compromised key(s) that have not been revoked/updated during a group communication interval (T_{GC}), representing the risk introduced by compromised keys (CK). This is computed by:

$$N_{CK}^{risk} = T_{GC}\left[\int_{0}^{LT} \sum_{i\in M, i\notin I} \sum_{j\in C, j\neq i} A_{i,j}(t)\right] \Big/ LT \qquad (8)$$

  where $A_{i,j}(t) = 1$ if node i has the correct public key of node j; 0 otherwise. I is the set of nodes that are considered trustworthy in integrity trust. C is the set of legitimate members whose private keys are compromised. M is the set of legitimate members in the network. We assume that if a node's private key is detected as compromised, it will not use the associated public key for group communication.

- **Number of Correct Public Keys ($\mathbf{N_{CPK}}$)**: This refers to the average number of correct public keys of other nodes kept in each node, computed by:

$$N_{CPK} = \left[\int_{0}^{LT} \frac{\sum_{i\in M}\sum_{j\in M, j\neq i} A_{i,j}(t)}{|M|}\right] \Big/ LT \qquad (9)$$

- **Correct Public Key Ratio ($\mathbf{P_{CPKR}}$)**: This refers to the fraction of correct public keys over the total number of public keys kept in each node, obtained by:

$$P_{CPKR} = \left[\int_{0}^{LT} \frac{\sum_{i\in M}\sum_{j\in M, j\neq i} A_{i,j}(t)}{\sum_{i\in M}\sum_{j\in M, j\neq i} PK_{i,j}(t)}\right] \Big/ LT \qquad (10)$$

  where $PK_{i,j}(t) = 1$ if node i has a public key of node j at time t; 0 otherwise. Here we note that $PK_{i,j}(t)$ covers the possibility of an incorrect public key of node j since an untrustworthy node may disseminate an invalid/incorrect public key.

- **Service Availability Ratio ($\mathbf{P_{SA}}$)**: This refers to the ratio of the average time period that a node's valid/correct public key is kept by other nodes over the entire session time, calculated by:

$$P_{SA} = \left[\int_{0}^{LT} \frac{\sum_{i\in M}\sum_{j\in M, j\neq i} A_{i,j}(t)}{\sum_{i\in M}\sum_{j\in M, j\neq i} 1}\right] \Big/ LT \qquad (11)$$

This metric implies how much time each node keeps another node's correct public key which has not expired during the entire session time.

- **Communication Overhead ($\mathbf{C_{total}}$)**: This is the number of hop messages per time unit (second) caused by the proposed protocol, computed by:

$$C_{total} = \left[\int_0^{LT} C_{total}(t)\right] \Big/ LT \tag{12}$$

where $C_{total}(t) = C_{TE}(t) + C_{KM}(t)$

$$C_{KM}(t) = C_{KI}(t) + C_{KD}(t) + C_{KR}(t)$$

$C_{TE}(t)$ is the number of hop messages caused by trust evaluation and $C_{KM}(t)$ is the number of hop messages caused by key management. $C_{KM}(t)$ consists of three cost components: key issuance ($C_{KI}(t)$), key distribution ($C_{KD}(t)$), and key revocation ($C_{KR}(t)$). Note that for $C_{TE}(t)$, each node will periodically (in every T_{update}) disseminate the trust values of its 1-hop neighbors to nodes located within the trust chain length (TC).

4. NUMERICAL ANALYSIS AND RESULTS

This section shows numerical results obtained from simulation. Our simulation is conducted using SMPL [13] in C, an event driven simulator. Table 1 gives the set of parameters and their values for defining the simulation environment. We use the optimal trust chain length (i.e., TC=4) identified in [6] for maximizing trust accuracy (or minimizing trust bias) for this environment with approximately 0.07% of trust bias over 7000 observations. Initial values of each trust property are seeded with a random variable selected from the range in [GB, 1] based on uniform distribution where GB is the lower bound. The trust values are also affected by network conditions and link unreliability in competence and the number of encountered entities in social contact. The initial estimated trust value at time t=0 is set to 0.5, implying ignorance. We report the impact of various trust thresholds on the performance metrics defined in Section 3.4 with varying network hostility.

Table 1: System parameters and default values

Parameter	Value	Parameter	Value
GB	0.5 – 0.9	α, γ	0.1, 0.95
T_{th}	0, 0.5 – 0.9	TC	4
T_{warmup}	30 min	T_{update}	5 min
LT	24 hrs	T_{GC}	5 min
$1/\lambda_{comp}$	3 hrs	R	150 m
N	100	$1/\lambda$, $1/\mu$	1 hr, 4 hrs
v_i	(0, 10] m/s	P_{fn}, P_{fp}	(0, 0.05]

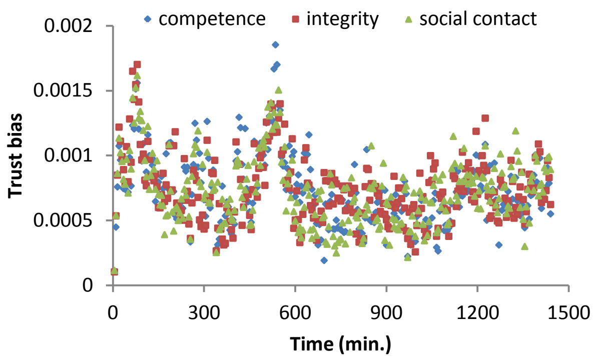

Fig. 1. Trust bias with (α, γ) = (0.1, 0.95) and TC = 4.

In Fig. 1, we show the trust bias observed with TC = 4 and (α, γ) = (0.1, 0.95) where α is a weight for direct evidence ((1- α) is a weight for indirect evidence) and γ is a decay factor used in Equation 1. We observe that α = 0.1, γ = 0.95 and TC = 4 are identified as optimal settings under which the trust bias is minimized at 0.2% as shown in Fig. 1. Note that each trust bias per time point in Fig. 1 is the mean of 25 observations. Accurate trust assessment is critical for a node to make effective decisions on whether to trust another node or not in order to reduce loss of opportunities due to underestimation or mitigate security vulnerability introduced by overestimation.

For Fig. 2 – Fig. 6, we show the results with standard deviations based on 100 simulation runs. We setup the attack intensity such that each node's private key can be compromised once per 3 hours ($1/\lambda_{comp}$) in this case study. We allow a 30 min. warm-up period (T_{warmup}) for peer-to-peer trust evaluation to reach a sufficiently accurate level. We use a 5 min. trust update interval (T_{update}).

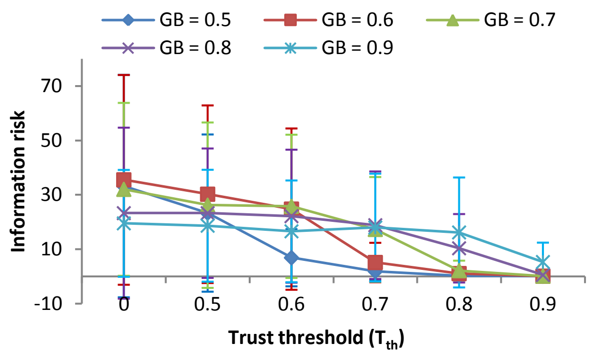

Fig. 2. Information Risk (N_{CK}^{risk}) vs. trust thresholds (T_{th}) under various initial trust levels (GB).

Recall that a node uses T_{th} to filter out trustworthy nodes to make decisions in key management. Using a lower T_{th} means that more nodes share confidential information while using a higher T_{th} implies fewer nodes know it. Fig. 2 shows the impact of the trust threshold (T_{th}) on information risk (N_{CK}^{risk}) as the initial trust level (GB as the lower bound of the seeds) varies. The case with zero trust threshold used (i.e., T_{th} =0) is a baseline model where no trust is being used in decision making. Overall, we observe that using a higher T_{th} mitigates information risk more because fewer nodes know the public key of a compromised private key. On the other hand, using a lower T_{th} increases the chance that many nodes share the public key of the compromised private key, resulting in a higher information risk. Particularly, when there are more trustworthy nodes (i.e., GB = 0.9), relatively more nodes know the public key even using a high T_{th} because more nodes exist with the trust values equal to or above T_{th}. Note that the standard deviation error bar is long especially when the trust threshold is low and GB is low because nodes with a wide range of trustworthiness can participate in key distribution.

Fig. 3 explains how T_{th} affects the average number of correct public keys kept by each node as GB varies. Intuitively, when there are more trustworthy nodes in the network (e.g., GB = 0.8 or 0.9), each node has more chances to obtain correct public keys of others because more nodes are qualified for the given T_{th} and fewer nodes drop the public key request message or provide incorrect public keys. In addition, using a lower T_{th} helps a node obtain more correct public keys due to the use of a more relaxed trust threshold, while using a higher T_{th} provides less chances for a node to obtain correct public keys as fewer nodes can access public keys.

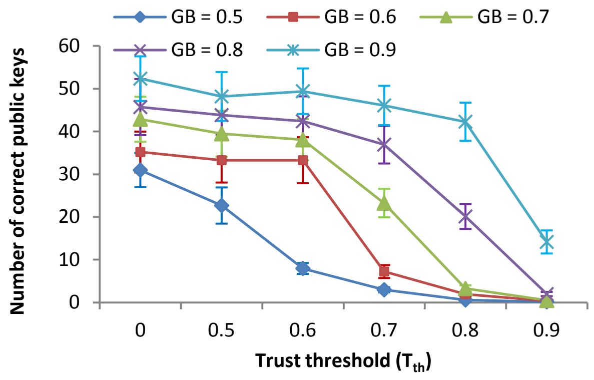

Fig. 3. Number of correct public keys (N_{CPK}) vs. trust thresholds (T_{th}) under various initial trust levels (GB).

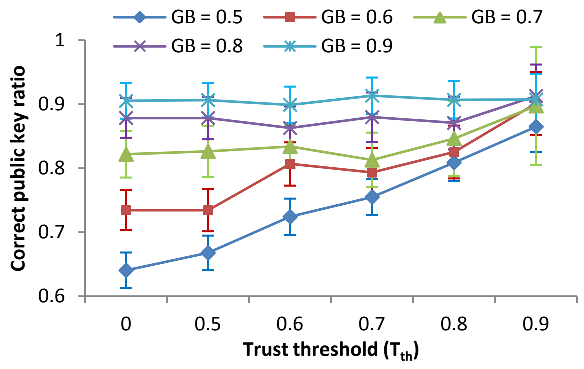

Fig. 4. Correct public key ratio (P_{CPKR}) vs. trust thresholds (T_{th}) under various initial trust levels (GB).

Fig. 4 shows the effect of T_{th} on the correct public key ratio (P_{CPKR}) as GB varies. Recall that P_{CPKR} is defined as the ratio of the average number of correct public keys to the total number of public keys (regardless of correctness) kept in each node. As shown in Fig. 4, the correctness of the public keys obtained increases under a higher T_{th}. This implies that when a higher T_{th} is used, there is a smaller chance for a public key to be modified. That is, as long as the public key of a node is safely delivered to the requestor, it is more likely to be authentic because both the provider of the public key and the intermediate nodes forwarding the public key are trustworthy using a higher T_{th}. Notice that when there are more trustworthy nodes (i.e., GB = 0.8 or 0.9), P_{CPKR} is almost insensitive to T_{th} because most nodes behave well regardless of the level of T_{th} used.

Fig. 5 addresses how T_{th} impacts service availability ratio (P_{SA}) as GB varies. Recall that P_{SA} indicates the average time that a node has a valid (not expired) and correct public key of other nodes over the entire session time. A node's public/private key pair is updated periodically so other nodes may keep an obsolete copy of the public key. In addition, a node may have an incorrect public key of another node because untrustworthy nodes may have modified the public key and provided it to the requesting node. We observe from Fig. 5 that except the case in which most nodes are trustworthy (i.e., GB = 0.9), there exists an optimal T_{th} that maximizes P_{SA}. The reason is that when a lower T_{th} is used, public keys are more likely to be modified or dropped because untrustworthy nodes have chances to forward or provide public keys upon request/distribution. On the other hand, when a higher T_{th} is used, the standard for forwarding/providing a public key is stricter and fewer nodes are qualified for providing the public key. Essentially the optimal T_{th} maximizing P_{SA} exists due to the inherent tradeoff between trust and risk. That is, using a high T_{th} is regarded as applying low trust towards other nodes (avoiding risk for security) while using a low T_{th} is interpreted as applying high trust towards other nodes (taking risk for performance).

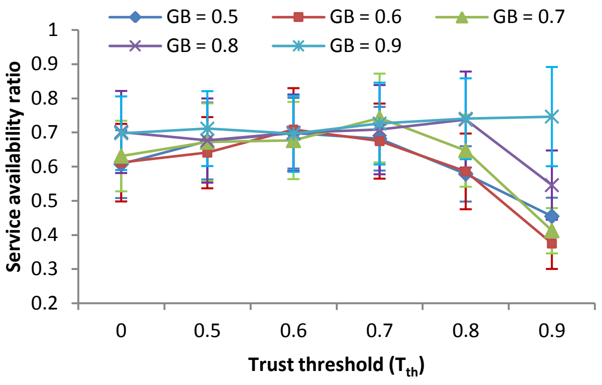

Fig. 5. Service availability ratio (P_{SA}) vs. trust thresholds (T_{th}) under various initial trust levels (GB).

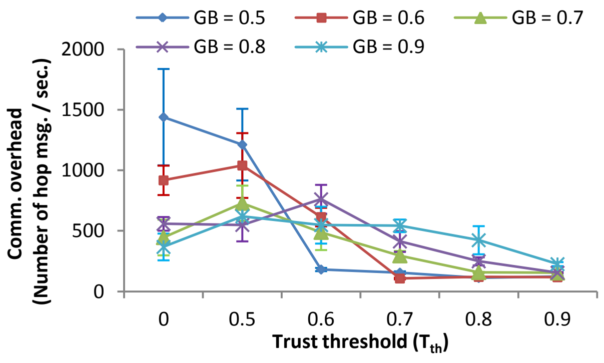

Fig. 6. Communication overhead (C_{total}) vs. trust thresholds (T_{th}) under various initial trust levels (GB).

Fig. 6 shows the impact of T_{th} on communication overhead (C_{total}) as GB varies. Recall that C_{total} increases when more requests for public keys are generated due to the detection of incorrect public keys kept in each node. In addition, C_{total} grows when more nodes are involved in forwarding messages. On the other hand, C_{total} decreases when a node obtains a correct public key of another node upon the first request/distribution. C_{total} also decreases when fewer nodes are involved in distributing public keys. When fewer trustworthy nodes are in the network (i.e., GB < 0.8), using a lower T_{th} increases C_{total} because fewer correct public keys are received in each node, thus introducing extra communication overhead to request the same public keys again.

In addition, using lower T_{th} allows more nodes to be involved in distributing public keys, leading to a higher communication overhead. By contrast, when a higher T_{th} is used, much fewer nodes are participating in distributing public keys, resulting in less communication overhead. Note that when $T_{th} = 0$, the communication cost due to trust evaluation is not counted because the case with $T_{th} = 0$ is a baseline scheme that does not use any trust in each node's decision making.

When more trustworthy nodes exist in the network (i.e., GB = 0.8 or 0.9), most nodes behave well regardless of the level of T_{th} used. Thus, we observe that C_{total} tends to be insensitive to T_{th} under trustworthy network environments.

5. CONCLUSION

In this paper, we proposed a composite trust based public key management scheme (CTPKM) for MANETs. Considering three different trust dimensions, namely, competence, integrity, and social contact, the proposed CTPKM enables a node to make decisions in interacting with others based on their trust levels. We devised five performance metrics to investigate the impact of the trust threshold (T_{th}) on security vulnerability (i.e., information risk and correct public key ratio metrics), availability (i.e., the number of correct public keys obtained in each node and service availability ratio metrics), and performance (i.e., communication overhead metric) in CTPKM.

We conclude the findings obtained from extensive simulation as follows. Using a higher T_{th} is preferred to minimize the information risk, correct public key ratio, and communication overhead metrics, while using a lower T_{th} is desirable to obtain more correct public keys. Further, there exists an optimal T_{th} that maximizes the service availability ratio metric due to the inherent tradeoff between trust and risk. Leveraging the tradeoff between trust and risk studied in this work, system designers may fine-tune the protocol setting to maximize performance while enhancing security, when given a set of parameter values as given in Table 1 characterizing the network environmental conditions.

In this work, we assumed random mobility. In the future, we plan to apply mobility traces in sparse or social MANETs as in [5] and analyze the effect of mobility on CTPKM performance. In this work, we also assumed a single threshold for trustworthiness classification. We plan to investigate more sophisticated fuzzy failure criteria as in [2] to further enhance CTPKM performance.

6. REFERENCES

[1] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," Workshop on Secure Mobile Ad-hoc Networks and Sensors, 2005.

[2] F. B. Bastani, I. R. Chen, and T. Tsao, "Reliability of systems with fuzzy-failure criterion," *Annual Reliability and Maintainability Symposium*, pp. 442-448, Anaheim, CA, USA, Jan. 1994.

[3] S. Capkun, L. Buttya, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52-64, Jan. – Mar., 2003.

[4] B.-J. Chang and S.-L. Kuo, "Markov chain trust model for trust-value analysis and key management in distributed multicast MANETs," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 1846-1863, May 2009.

[5] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Trust management for encounter-based routing in delay tolerant networks," *IEEE Global Communications Conf.*, pp. 1-6, Miami, Florida, USA, Dec. 2010.

[6] J.H. Cho, A. Swami, and I.R. Chen, "Modeling and Analysis of Trust Management with Trust Chain Optimization in Mobile Ad Hoc Networks," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1001-1012, May 2012.

[7] E. Da Silva, A. Dos Santos, L.C.P. Albini, and M. Lima, "Identity-based key management in mobile ad hoc networks: techniques and applications," *IEEE Wireless Communications*, vol. 15, no. 5, pp. 46-52, Oct. 2008.

[8] H. Dahshan, and J. Irvine, "A trust based threshold cryptography key management for mobile ad hoc networks," *IEEE 70th Vehicular Technology Conf.*, pp. 1-5, Anchorage, AK, USA, Sept. 2009.

[9] Y. G. Desmedt, "Threshold cryptography," European Transactions on Telecommunications, vol. 5, no. 4, pp. 449-458, July/Aug. 1994.

[10] H. Huang and S. F. Wu, "An approach to certificate path discovery in mobile ad hoc networks," *1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, Oct. 2003.

[11] J. Huang and D. Nicol, "A calculus of trust and its application to PKI and identity management," *ACM 8th Symposium on Identity and Trust on the Internet*, Gaithersburg, MD, USA, April 2009.

[12] X. Li, X. Wang, and J. Shen, "Strategy and simulation of trust cluster based key management protocol for ad hoc networks," *4th Int'l Conf. on Computer Science & Education*, pp. 269 - 274, Nanning, China, July 2009.

[13] M. H. MacDougall, *Simulating Computer Systems*, Computer Systems Series, The MIT Press, 1987.

[14] S. Sattam, A. Riyami, and K. G. Paterson, "Certificateless public key cryptography," *9th Int'l Conf. on the Theory and Application of Cryptology and Information Security*, Taipei, Taiwan, Nov. 2003, Advances on Cryptology, Lecture Notes in Computer Science, vol. 2894, pp. 452-473, 2003.

[15] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *CRYPTO'84*, 1984, pp. 47–53.

[16] B. Solhaug, D. Elgesem, and K. Stolen, "Why trust is not proportional to risk?" *2nd Int'l Conf. on Availability, Reliability, and Security*, April 2007, Vienna, Austria, pp. 11-18.

[17] H. Sun, X. Zheng, and Z. Deng, "An identity-based and threshold key management scheme for ad hoc networks," *Int'l Conf. on Networks Security, Wireless Communications and Trusted Computing*, pp. 520-523, Wuhan, Hubei, China, April 2009.

[18] N. V. Vinh, M.-K. Kim, H. Jun, and N. Q. Tung, "Group-based public-key management for self-securing large mobile ad-hoc networks," *Int'l Forum on Strategic Technology*, pp. 250-253, Oct. 2007.