Securing Mobile Ad Hoc Networks with Certificateless Public Keys

Yanchao Zhang, *Member*, *IEEE*, Wei Liu, Wenjing Lou, *Member*, *IEEE*, and Yuguang Fang, *Senior Member*, *IEEE*

Abstract—This paper studies key management, a fundamental problem in securing mobile ad hoc networks (MANETs). We present IKM, an ID-based key management scheme as a novel combination of ID-based and threshold cryptography. IKM is a certificateless solution in that public keys of mobile nodes are directly derivable from their known IDs plus some common information. It thus eliminates the need for certificate-based authenticated public-key distribution indispensable in conventional public-key management schemes. IKM features a novel construction method of ID-based public/private keys, which not only ensures high-level tolerance to node compromise, but also enables efficient network-wide key update via a single broadcast message. We also provide general guidelines about how to choose the secret-sharing parameters used with threshold cryptography to meet desirable levels of security and robustness. The advantages of IKM over conventional certificate-based solutions are justified through extensive simulations. Since most MANET security mechanisms thus far involve the heavy use of certificates, we believe that our findings open a new avenue towards more effective and efficient security design for MANETs.

Index Terms—Mobile ad hoc networks, security, key management, ID-based cryptography, secret sharing.

1 Introduction

Mobile ad hoc networks (MANETs) are infrastructureless, autonomous, stand-alone wireless networks that are receiving growing attention from both academia and industry. Security support is indispensable for typical application scenarios of MANETs such as military and homeland security operations. Security design for MANETs is, however, complicated by a number of unique features of MANETs. Of note are the lack of infrastructure, shared wireless medium, node mobility, resource constraints of mobile devices, bandwidth-limited and error-prone channels, and so on [1]. In this paper, we are concerned with key management, the foundation on which to build any other security mechanism for MANETs.

Conventional key management techniques may either require an online trusted server or not. The infrastructureless nature of MANETs precludes the use of server-based protocols such as Kerberos [2]. We therefore focus on discussing serverless approaches from here on. There are two intuitive symmetric-key solutions, though neither is satisfactory. The first one is to preload all the nodes with a global symmetric key, which is vulnerable to any point of compromise: If any single node is compromised, the security of the entire network is breached. Assuming a network of N nodes, the other solution is to let each pair of nodes maintain a unique secret that is only known to those two nodes. This approach suffers from three main drawbacks making it also unsuitable for MANETs. First, it lacks scalability because it is difficult to establish pairwise symmetric keys between existing nodes and newly joined nodes. Second, securely updating the overall $N(N-1)/2$ keys in the network is a nontrivial (if not impossible) task, as the size of the network increases. Last, it requires each node to store $(N-1)$ keys, which may represent a significant storage overhead in a large network. Symmetric-key techniques are also commonly criticized for not supporting efficient digital signatures because each key is known to at least two nodes. This renders public-key solutions more appealing for MANETs, which are the theme of this paper.

There has been a rich literature on public-key management in MANETs, see [3], [4], 5], [6], [7], [8] for example. These schemes all depend on certificate-based cryptography (CBC), which uses public-key certificates to authenticate public keys by binding public keys to the owners' identities. A main concern with CBC-based approaches is the need for certificate-based public-key distribution. One naive method is to preload each node with all the others' public-key certificates prior to network deployment. This approach can neither scale well with the increasing network size, nor handle key update in a secure and cost-effective way. Another approach of on-demand certificate retrieval may cause both unfavorable communication latency and often tremendous communication overhead, which will be justified via simulations in Section 5.5.

As a powerful alternative to CBC, ID-based cryptography (IBC) [9] has been gaining momentum in recent years.

- *Y. Zhang is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102. E-mail: yczhang@njit.edu.*
- *W. Liu is with Scalable Network Technologies, Los Angeles, CA 90045. E-mail: liuw@ufl.edu.*
- *W. Lou is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609. E-mail: wjlou@ece.wpi.edu.*
- *Y. Fang is with the Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building, PO Box 116130, Gainesville, FL 32611. E-mail: fang@ece.ufl.edu.*

Manuscript received 16 June 2005; revised 20 Apr. 2006; accepted 17 May 2006; published online 2 Nov. 2006.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0091-0605.

TABLE 1
Notation

p, q	two large primes	$\mathbb{G}_1, \mathbb{G}_2$	cyclic groups of order q
$\hat{e}$	pairing s.t. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$	H_1	mapping strings to non-zero elements in $\mathbb{G}_1$
Ψ	the network node set, $\lvert\Psi\rvert = N$	Ω	the D-PKG set, $\lvert\Omega\rvert = n$
ID_A	network ID of node A	t, n	secret-sharing parameters
$g(x)$	$(t-1)$-degree polynomial	$\lambda_V(x)$-s	Lagrange coefficients
$\overline{ID_A}$	key revocation against node A	$K_{\mathsf{P1}}, K_{\mathsf{P2}}$	two distinct network master secrets
W	generator of $\mathbb{G}_1$	$W_{\mathsf{P1}}, W_{\mathsf{P2}}$	$W_{\mathsf{P1}} = K_{\mathsf{P1}}W \in \mathbb{G}_1$, $W_{\mathsf{P2}} = K_{\mathsf{P2}}W \in \mathbb{G}_1$
$k_{A,B}$	symmetric key shared between A and B	p_i	i^{th} key update period, for $1 \leq i \leq M$
$\mathcal{K}_A/\mathcal{K}_A^{-1}$	node-specific public-key and private-key elements of node A	$\mathcal{K}_{p_i}/\mathcal{K}_{p_i}^{-1}$	common public-key and private-key elements in phase p_i
salt_i	unique binary string associated with p_i	$\mathcal{K}_{A,p_i}/\mathcal{K}_{A,p_i}^{-1}$	public/private keys of node A in phase p_i
K_{P2}^V	the D-PKG V's secret share of K_{P2}	γ	revocation threshold
$\mathcal{F}$	mapping a given node ID to β D-PKG IDs	h	hash function such as SHA-1 [16]
$\{m\}_{k_x}$	message m encrypted under key k_x with a symmetric-key primitive	$[m]_{\mathcal{K}_{A,p_i}^{-1}}$	message m with its ID-based signature generated under private key $\mathcal{K}_{A,p_i}^{-1}$

It allows public keys to be derived from entities' known identity information, thus eliminating the need for public-key distribution and certificates. This nice feature has inspired a few IBC-based certificateless public-key management schemes for MANETs such as [10], [11], [12], [13]. The basic idea is to let some [10], [11], [13] or all network nodes [12], called *shareholders*, share a network master-key using threshold cryptography [14], [15] and collaboratively issue ID-based private keys. There, however, remain many issues to be satisfactorily resolved. First of all, the security of the whole network is breached when a threshold number of shareholders are compromised. Second, updating ID-based public/private keys requires each node to individually contact a threshold number of shareholders, which represents a significant communication overhead in a large-scale MANET. Third, except our preliminary result in [13], none of the existing proposals consider how to select the secret-sharing parameters used with threshold cryptography to achieve desirable levels of security and robustness. Last, there is no comprehensive quantitative argument about the advantages of IBC-based public-key management schemes over CBC-based ones.

In this paper, we address all the above concerns by devising an ID-based key management scheme, called ***IKM***, for special-purpose MANETs administered by a single authority. MANETs of this type have long been recognized and will continue to be one of the major application categories of wireless ad hoc networking techniques. Typical examples are those deployed in military battlefield operations and homeland security scenarios. Our major contributions are as follows:

- **A novel construction method of ID-based public/private keys**. In IKM, each node's public key as well as private key is composed of a node-specific, ID-based element and a network-wide common element. Node-specific key elements ensure that the compromise of arbitrarily many nodes does not jeopardize the secrecy of noncompromised nodes' private keys; common key elements enable very efficient network-wide public/private key updates via a single broadcast message. We also discuss efficient key agreement, public-key encryption, and digital signatures based on such public/private keys.
- **Determining secret-sharing parameters used with threshold cryptography**. Similarly to [10], [11], [12], we apply threshold cryptography to distribute a network master-key among some shareholders. Different from them, we identify devastating pinpoint attacks against shareholders and propose the corresponding countermeasure based on anonymous routing [17]. In addition, we discuss how to choose the secret-sharing parameters for meeting desirable levels of security and robustness.
- **Simulation studies of advantages of IKM over CBC-based schemes**. By detailed simulations, we show that IKM has performance equivalent to CBC-based schemes, denoted by *CKM*, with regard to key revocation, while it behaves much better in key updates. Furthermore, we demonstrate that IKM is able to turn an elegant CKM-based secure routing protocol [18] into a much more efficient one.

Since most existing MANET security mechanisms rely on the heavy use of certificates, we believe that our findings open a new avenue towards more effective, efficient security designs.

The rest of the paper is organized as follows: In Section 2, we survey the related work and outline a *pairing* technique. Next, we present design goals and the network and adversary models in Section 3, followed by a detailed illustration of the IKM design in Section 4. Then, the simulation-based comparative study of our IKM and CKM is given in Section 5, and this paper is finally concluded in Section 6.

2 Preliminaries

In this section, we first define the notation to be used in the rest of this paper. We then survey the related work and outline the *pairing* technique on which we base our design.

2.1 Notation

For clarity, Table 1 lists some important notation whose concrete meanings will be further explained where they appear for the first time.

2.2 Related Work

Due to space limitations, we only discuss prior art that is more germane to our work, and refer to [1] for a more comprehensive survey.

The seminal paper by Zhou and Hass [3] suggests using CBC and (t, n)-threshold cryptography [14], [15] in MANETs. Let N be the overall number of nodes and t, n be two integers satisfying $t \leq n > N$. In [3], prior to network deployment, the CA's public key is furnished to each node, while its private key is divided into n shares, each uniquely assigned to one of n chosen nodes called *D-CA*s hereafter. During network operation, any t D-CAs can jointly perform certificate generation and revocation based on their secret shares, while any less than t D-CAs cannot. Yi and Kravets [6] proposes selecting computationally more powerful and physically more secure nodes as D-CAs. Both schemes can tolerate the compromise of up to $(t-1)$ D-CAs so that adversaries cannot reconstruct the CA's private key, and the failure of up to $(n-t)$ D-CAs so that there are always at least t functional D-CAs.

Different from [3], [6], URSA [4], [8] is a (t, N)-threshold scheme in which each of the N nodes is a D-CA. The advantage of URSA is the increased service availability in that a certificate can now be generated or revoked by any t nearby nodes, and URSA can tolerate the failure of up to $(N-t)$ D-CAs. The disadvantage, however, is that the compromise of any t out of N nodes would expose the CA's private key and thus result in loss of overall system security [6]. In addition, as noted in [19], URSA is vulnerable to the Sybil attack [20] because an adversary can take as many identities as necessary to collect enough shares and reconstruct the CA's private key. Other security problems of URSA are analyzed in [5], [21].

All the above schemes are based on RSA [22], either explicitly [4], [8] or implicitly [3], [6], [7]. By comparison, the scheme [5] relies on DSA [23] and threshold cryptography, and has much worse communication efficiency than RSA-based schemes. The reason is that, to tolerate the compromise of up to $(t-1)$ D-CAs, the DSA-based scheme needs to contact $(2t-1)$ D-CAs for generating a new certificate, while RSA-based approaches only involve t D-CAs [5]. Please refer to [12] for simulation studies of the communication inefficiency of DSA-based approaches.

The aforementioned CBC-based schemes are all targeted for single-authority MANETs as what we have in mind. Another notable line of approaches such as [19], [24] is to let each node act as a CA to issue certificates to other nodes. While maybe suitable for authority-less civilian networks, they are less fit for single-authority MANETs under consideration.

Despite its attractive features, IBC has not received deserved attention as a powerful tool to secure MANETs until recently. Khalili et al. [10] suggest using IBC and threshold cryptography in MANETs, but their work is conceptual. Deng et al. [11] present an ID-based key management scheme for authority-less MANETs, so it is less applicable to single-authority MANETs we aim at. Bohio and Miri [25] propose to use ID-based keys for secure broadcast, but their work is not intended for efficient key management. Our preliminary work [13] also addresses the secure application of IBC to MANETs. In addition, Zhang et al. develop MASK [17], [26], an IBC-based anonymous on-demand routing protocol for MANETs.

The closest work to ours is ID-GAC [12], in which Saxena et al. present an elegant IBC-based access control scheme for ad hoc groups such as MANETs. ID-GAC is basically a (t, N)-threshold scheme, in which, prior to deployment, each of the N nodes is furnished with a share of a master-key. Although having high-level service availability as URSA [8], ID-GAC suffers from the same undesirable security drawback mentioned above. In contrast, our IKM is a (t, n)-threshold scheme, similar to [3], [6]. At first glance, IKM is less robust than ID-GAC because it only tolerates the failure of up to $(n-t)$ shareholders instead of $(N-t)$ in ID-GAC. However, this also means that IKM is more secure than ID-GAC because the fewer shareholders make it feasible to spend more in safeguarding them, for instance, by enclosing them in high-quality tamper-resistant devices and/or putting them under better monitoring. In addition, our IKM incorporates an additional defense line by making shareholders indistinguishable from common nodes via anonymous routing [17]. Furthermore, even when t or more shareholders are compromised and the master-key is exposed, our novel public/private key construction method guarantees that private keys of noncompromised nodes remain safe. This is in contrast to the overall loss of security in ID-GAC (see Section 4.7). Moreover, each noncompromised node in ID-GAC needs to individually contact t shareholders for key update. In contrast, our IKM is much more efficient in both computation and communication by updating public/private keys of all the noncompromised nodes via a single broadcast message. As an addition, ID-GAC suffers from the Sybil attack as URSA, while our IKM does not.

2.3 Pairing Technique

Although the idea of IBC dates back to 1984 [9], only recently has its rapid development taken place due to the application of the *pairing* technique outlined below.

Let p, q be two large primes[1] and $\mathrm{E}/\mathbb{F}_p$ indicate an elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. We denote by $\mathbb{G}_1$ a q-order subgroup of the additive group of points of $\mathrm{E}/\mathbb{F}_p$, and by $\mathbb{G}_2$ a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. The Discrete Logarithm Problem (DLP) is required to be hard[2] in both $\mathbb{G}_1$ and $\mathbb{G}_2$. For us, a pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. *Bilinear*: $\forall\ P, Q, R, S \in \mathbb{G}_1$,

$$\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S). \tag{1}$$

Consequently, for $\forall\ a, b \in \mathbb{Z}^*_q$, we have

$$\hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}, \text{etc.}$$

1. The conditions that p, q must satisfy are given in [27], [28].

2. It is computationally infeasible to extract the integer $x \in \mathbb{Z}^*_q = \{a | 1 \leq a \leq q-1\}$, given $P, Q \in \mathbb{G}_1$ (respectively, $P, Q \in \mathbb{G}_2$) such that $Q = xP$ (respectively, $Q = P^x$).

2. *Nondegenerate*: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P,P) \in \mathbb{F}^*_{p^2}$ is a generator of $\mathbb{G}_2$.
3. *Computable*: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P,Q \in \mathbb{G}_1$.

Note that $\hat{e}$ is also *symmetric*, i.e., $\hat{e}(P,Q) = \hat{e}(Q,P)$, for all $P,Q \in \mathbb{G}_1$, which follows immediately from the bilinearity and the fact that $\mathbb{G}_1$ is a cyclic group. Modified Weil [27], [28] and Tate [29] pairings are examples of such bilinear maps for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard.[3] We refer to [27], [28], [29] for a more comprehensive description of how these pairing parameters should be selected in practice for efficiency and security.

3 Design Goals and System Models

In this section, we present our design goals as well as network and adversary models.

3.1 Design Goals

From our point of view, a sound key management scheme for MANETs should satisfy the following requirements. First, it must not have a single point of compromise and failure because mobile nodes deployed in hostile environments are subject to either logical or physical attacks. Second, it should be *compromise-tolerant*, meaning that the compromise of a certain number of nodes does not harm the communication security between noncompromised nodes. Third, it should be able to efficiently and securely revoke keys of compromised nodes once detected and update keys of noncompromised nodes. Last, it should be efficient in terms of storage, computation, and communication, as mobile nodes are usually very resource-constrained. It is worth stressing that communication efficiency is far more important an issue in MANETs than in wireline networks, as wireless transmission of a bit can require over 1,000 times more energy than a single 32-bit computation (see [30]). We thus must seek ways to reduce communications related to key management as much as possible.

3.2 Network Model

We consider a special-purpose, single-authority MANET consisting of N nodes, denoted by a set notation $\Psi(|\Psi| = N)$. The network size N may be dynamically changing with node join, leave, or failure over time. Depending on different applications, N may range from several tens to several thousands or even more. Each node $A \in \Psi$ has a unique ID, denoted by ID_A and assumed to be its network-layer address as usual.

We assume that each node has limited transmission and reception capabilities. Two nodes out of transmission range of each other can communicate via a sequence of intermediate nodes in a multihop fashion. Since all the nodes belong to a single authority and, thus, have common interests, node selfishness [31] is not worrysome in that each node is ready to forward packets not destined for itself. Nodes may freely move in the network, but do not continuously move so rapidly as to make the flooding of every data packet the only feasible routing protocol. This is a common assumption made about node mobility by nearly all MANET schemes. We further assume that nodes are capable of performing public-key operations, which is reasonable for the targeted application scenarios, though symmetric-key operations should be used instead whenever possible.

Our IKM is independent of the underlying transport, routing, or MAC protocols. However, we do assume that, whenever needed, a valid unicast route can be established between any two nodes. This can be achieved through many existing secure routing protocols, such as ARAN [18]. It is worth pointing out that, similar to almost all the other existing secure routing schemes, ARAN is built upon conventional certificates. In Section 5.5, we will show that it can be easily converted into a much more efficient scheme based on our IKM.

3.3 Adversary Model

Our intention here is to devise a sound key management scheme for MANETs, so we just consider attacks aimed at key management itself. Mitigating denial-of-service attacks, such as physical-layer jamming, MAC-layer misbehavior, or routing disruption, though important, is beyond the scope of this paper.

Attacks can be mounted by a single adversary or collaborative ones. We differentiate between node *compromise* and *disruption* attacks. By saying that a node is compromised, we mean that adversaries have complete control over it, including learning or modifying its secret information, changing its intended behavior, and so on. In contrast, disrupting a node means that adversaries can only disrupt communication to that node, e.g., by interfering with wireless signals to and from it, but cannot read the secret information stored on it. Therefore, node disruption attacks are less severe than node compromise attacks. However, we assume that adversaries cannot compromise or disrupt an unlimited number of nodes so that legitimate nodes are always the majority. Nor can they break any of the cryptographic primitives on which we base our design. In addition, we assume *static* instead of *dynamic* adversaries [32].

We further assume that compromised nodes will eventually exhibit detectable misbehavior. There is unlikely to be a valid security solution if compromised nodes remain "passive." As [4], [8], we assume an efficient misbehavior detection scheme such as [33] or [34]. One of our main objectives is to drive identified compromised nodes out of the network by revoking their keys. Hereafter, we use *compromised nodes* to indicate those which have been compromised and identified, unless otherwise stated.

There are n distributed authorities called *D-PKGs* in our IKM, similar in role to the distributed CAs (D-CAs) in conventional CKM [3], [4], [5], [6], [7], [8]. The D-PKGs differ from common nodes only in that each of them knows a share of a network master-secret. Similarly to [3], [4], [5], [6], [7], [8], our IKM works properly on the assumption that adversaries can compromise at most $(t-1)$ D-PKGs and can disrupt no more than $(n-t)$ D-PKGs. For the sake of simplicity, we refer to this assumption as the $t\text{-}limited$ assumption. Note that this $t\text{-}limited$ assumption only needs to hold in each predetermined time period rather than the whole network lifetime, if proactive secret sharing [35] is used to periodically refresh secret shares of the D-PKGs.

3. It is believed that, given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}^*_q$ and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P,P)^{xyz} \in \mathbb{G}_2$ with nonnegligible probability.

4 IKM Design

This section presents our IKM design. We first provide an overview of IKM in Section 4.1, and then describe the key predistribution phase in Section 4.2. Next, we discuss how to achieve efficient key revocation and update in Sections 4.3 and 4.4, respectively. Section 4.5 presents our method of protecting the D-PKGs from devastating pinpoint attacks, and Section 4.6 gives general guidelines as to how to select the secret-sharing parameters t, n. Finally, the security of IKM is analyzed in Section 4.7.

4.1 Overview

In IKM, each node should carry an authentic ID-based public/private key pair at any time as a proof of its group membership. With such key pairs, nodes can realize mutual authentication, key agreement, public-key encryption, and digital signatures, among other security services. IKM consists of three phases: key predistribution, revocation, and update.

Key predistribution is a one-time process occurring during network initialization, where a Private Key Generator (PKG), essentially a trusted authority, determines a set of system parameters and preloads every node with appropriate keying materials. In addition, the PKG distributes its functionality to n D-PKGs selected among the N nodes to enable secure and robust key revocation and update during network operation.

To minimize the damage from node compromise, it is a must to explicitly revoke public keys of compromised nodes. During network operation, if suspecting that a peer, say A, has been compromised, a node sends a signed accusation against A to some D-PKGs. The accused A is diagnosed as compromised when the number of accusations against it reaches a predefined *revocation threshold*, denoted by γ, in a certain time window. At that point, the network enters the key revocation phase in which the D-PKGs jointly issue a key revocation against A.

As a common practice [8], public/private keys of mobile nodes need to be updated at intervals for many reasons, e.g., preventing from cryptanalysis. The key update phase may occur either periodically according to a prescribed time period, or reactively when the number of revoked nodes attains some predetermined threshold. During this phase, each nonrevoked node can update its public key autonomously and its private key via a single broadcast message. This is enabled by our novel public/private key construction method. Our scheme can also ensure that compromised nodes, once revoked, cannot get their keys updated, thus isolated from the network.

Due to the shared wireless medium, adversaries are easy to find the whereabouts of D-PKGs based on their network IDs leaked in routing and data packets [17]. This renders the D-PKGs particularly vulnerable to devastating pinpoint attacks. As a natural defense, we propose to make the D-PKGs indistinguishable from common nodes via anonymous routing [17]. This measure allows us to provide general guidelines about how to choose the secret-sharing parameters t, n for achieving desirable levels of security and robustness.

4.2 Network Initialization

For a single-authority MANET under consideration, it is reasonable to assume a trusted PKG will bootstrap the network, which itself is not part of the resulting network.

4.2.1 Generation of Pairing Parameters

To bootstrap the network, the PKG does the following:

1. Generate the pairing parameters $(p, q, \hat{e})$, as described in Section 2.3. Select an arbitrary generator W of $\mathbb{G}_1$.
2. Choose a hash function[4] H_1 that maps arbitrary binary strings to nonzero elements in $\mathbb{G}_1$.
3. Pick two distinct random numbers K_{P1}, $K_{\mathrm{P2}} \in \mathbb{Z}_q^*$ as network master-secrets. Set $W_{\mathrm{P1}} = K_{\mathrm{P1}}W$ and $W_{\mathrm{P2}} = K_{\mathrm{P2}}W$, respectively.

Parameters $(p, q, \hat{e}, H_1, W, W_{\mathrm{P1}}, W_{\mathrm{P2}})$ are public knowledge preloaded to each node, while K_{P1} and K_{P2} should never be disclosed to any single node.

4.2.2 Secret Sharing

To enable key revocation and update during network operation, it is necessary to introduce the PKG functionality into the network. In our design, only knowledge of K_{P2} is introduced into the network to ensure high-level compromise tolerance (analyzed in Section 4.7). To avoid the single point of compromise and failure, the PKG performs a (t, n)-threshold secret sharing of K_{P2} by first determining a random polynomial, $g(x) = K_{\mathrm{P2}} + \sum_{i=1}^{t-1} g_i x^i \pmod q$. It then randomly selects a subset $\Omega \subset \Psi$ of size n of nodes as D-PKGs $(t \le n < |\Psi| = N)$. Then, the PKG assigns to each $V \in \Omega$ a secret share computed as $K_{\mathrm{P2}}^V = g(ID_V)$. Based on Lagrange interpolation, any subset $\mathcal{A} \subset \Omega$ of size t can codetermine the polynomial:

$$g(x) = \sum_{V \in \mathcal{A}} \lambda_V(x) K_{\mathrm{P2}}^V \pmod q, \tag{2}$$

where $\lambda_V(x) = \prod_{S \in \mathcal{A} \setminus \{V\}} \frac{ID_S - x}{ID_S - ID_V}$ is called a Lagrange coefficient. The PKG's master secret K_{P2} can then be reconstructed by computing $g(0)$. However, any subset of Ω of size $(t-1)$ or smaller does not suffice to do so. To enable verifiable secret sharing, the PKG also calculates a set of values $\{W_{\mathrm{P2}}^V = K_{\mathrm{P2}}^V W | V \in \Omega\}$ preloaded to each D-PKG. Due to the difficulty in solving the DLP in $\mathbb{G}_1$, all the other D-PKGs cannot deduce the secret share K_{P2}^V of D-PKG V from W_{P2}^V. The IDs of all the D-PKGs are known to each node to make key revocation and update feasible, and the choice of t, n will be discussed in Section 4.6.

4.2.3 Generation of ID-Based Public/Private Keys

One of our essential design points is how to construct an ID-based public/private key pair for each node A, be it a D-PKG or common node. Our IKM is composed of a number of continuous, nonoverlapping *key update phases*, denoted by p_i for $1 \le i < M$, where M is the maximum

4. We assume that all the hash functions including H_1 used in this paper act like random oracles [36].

possible phase index. Such p_is may not be of the same length in time and, thus, do not require nodes to be time-synchronized for them either. Each p_i is associated with a unique binary string, called a *phase salt* and denoted by salt_i. Prior to deployment, the PKG issues a random number salt_1 to each node which, in turn, can subsequently generate $\text{salt}_i = \text{salt}_{i-1} + 1 (1 < i \leq M)$ by itself with an efficient hash function h such as SHA-1 [16].

In IKM, each public/private key pair is both *node-specific* and *phase-specific* and node A's key pair valid only during phase p_i is denoted by $< \mathcal{K}_{A,p_i}, \mathcal{K}_{A,p_i}^{-1} >$. Each of $\mathcal{K}_{A,p_i}$ and $\mathcal{K}_{A,p_i}^{-1}$ is comprised of a node-specific element and a phase-specific element common to all the nodes, both in $\mathbb{G}_1$. In particular,

$$\begin{cases} \mathcal{K}_{A,p_i} := (\mathcal{K}_A, \mathcal{K}_{p_i}) \ = (H_1(ID_A), H_1(\text{salt}_i)) \\ \mathcal{K}_{A,p_i}^{-1} := (\mathcal{K}_A^{-1}, \mathcal{K}_{p_i}^{-1}) = (K_{\text{P1}} H_1(ID_A), K_{\text{P2}} H_1(\text{salt}_i)). \end{cases}$$

Initially, the PKG issues $< \mathcal{K}_{A,p_1}, \mathcal{K}_{A,p_1}^{-1} >$ to node A which can acquire $< \mathcal{K}_{A,p_i}, \mathcal{K}_{A,p_i}^{-1} > \ (1 < i \leq M)$ from the D-PKGs during network operation, as will be shown later. For convenience, hereafter we refer to $< \mathcal{K}_{p_i}, \mathcal{K}_{p_i}^{-1} >$ as common public-key and private-key elements of phase p_i, and $< \mathcal{K}_A, \mathcal{K}_A^{-1} >$ as node-specific public-key and private-key elements of node A. The former pair varies across key-update phases, while the later pair remains unchanged during network lifetime and should be kept confidential to A itself.

Due to the difficulty of solving the DLP in $\mathbb{G}_1$, it is computationally infeasible to derive the network master-secrets K_{P1} and K_{P2} from an arbitrary number of public/private key pairs [27], [28]. It means that, no matter how many key pairs adversaries acquire from compromised nodes, they cannot deduce the private key of any non-compromised node. Therefore, our IKM exhibits the desirable compromise-tolerant property. The advantage of our key construction method in facilitating key update can be seen in Section 4.4. In addition, the resulting higher-level resilience to the compromise of D-PKGs than the conventional key construction method [12], [13] is to be analyzed in Section 4.7. Furthermore, we refer to the readers to [37] for the use of such public/private keys in key agreement, key agreement, encryption/decryption, and signature generation/verification.

Our IKM allows dynamic node join at any time and thus ensures high network scalability. Suppose a new node X joins the network at phase p_i. The PKG just needs to pre-equip X with public system parameters and $< \mathcal{K}_{X,p_i}, \mathcal{K}_{X,p_i}^{-1} >$.

4.2.4 Generation of Key-Update Parameters

Let t^c be the maximum number of compromised nodes the network can tolerate. To realize broadcast-based public/private key updates, the PKG picks M distinct $2t^c$-degree polynomials, denoted by $\{l_i(x) = \sum_{j=0}^{2t^c} l_{i,j} x^j (\text{mod } q)\}_{i=1,\ldots,M}$ with $l_{i,j} \in \mathbb{Z}_q^*$, and M distinct t^c-degree polynomials, denoted by

$$\{u_i(x) = \sum_{j=0}^{t^c} u_{i,j} x^j (\text{mod } q)\}_{i=1,\ldots,M}$$

with $u_{i,j} \in \mathbb{Z}_q^*$. Since $\mathcal{K}_{p_i}^{-1}$ is a point on $\text{E}/\mathbb{F}_p$, its x-coordinate (denoted as $[\mathcal{K}_{p_i}^{-1}]^x$) can be uniquely determined from its y-coordinate (denoted as $[\mathcal{K}_{p_i}^{-1}]^y$). The PKG then constructs $\{v_i(x) = [\mathcal{K}_{p_i}^{-1}]^y - u_i(x)\}_{i=1,\ldots,M}$, which are given to each node A along with $\{l_i(ID_A)\}_{i=1,\ldots,M}$.

4.2.5 Summary

To summarize, each node has the following cryptographic materials before network deployment:

- **Pairing parameters:** $(p, q, \hat{e}, H_1, W, W_{\text{P1}}, W_{\text{P2}})$.
- **Public and private keys:** $< \mathcal{K}_{A,p_1}, \mathcal{K}_{A,p_1}^{-1} >$.
- **Phase salt:** salt_1.
- **Key-update parameters:** $\{v_i(x), l_i(ID_A)\}_{i=1,\ldots,M}$.

In addition to the above materials, each D-PKG $V \in \Omega$ holds a secret share K_{P2}^V and values $\{W_{\text{P2}}^V = K_{\text{P2}}^V W | V \in \Omega\}$.

4.3 Key Revocation

Key revocation is comprised of three subprocesses: *misbehavior notification*, *revocation generation*, and *revocation verification*. The following description applies to phase p_i.

4.3.1 Misbehavior Notification

Upon detection of node A's misbehavior, node B generates a signed accusation $[ID_A, s_B]_{\mathcal{K}_{B,p_i}^{-1}}$ against A, where s_B is a timestamp for withstanding message replay attacks. The revocation needs to be sent to the D-PKGs to report A's misbehavior. The naive flooding of the accusation is insecure because it may *alert* the accused A to temporarily behave normally. By doing so, it attempts to make the number of accusations against it below the predefined revocation threshold γ to avoid being revoked. Therefore, B should unicast the accusation secretly to the D-PKGs. The next question is to which D-PKGs the accusation is sent. The following approach is adopted in IKM.

During network initialization, the PKG furnishes each node with a function $\mathcal{F}$ that maps each node ID to the IDs of β distinct D-PKGs. More formally, for node $A \in \Psi$, $\mathcal{F}(ID_A) = \{ID_{X_j} | 1 \leq j \leq \beta, X_j \in \Omega, X_j \neq A\}$. There are many possible ways to construct such a function. One simple approach is to divide the node set Ψ into n disjoint node sets, each associated with β D-PKGs. However, the condition that must be satisfied is that the node set a D-PKG belongs to should not be associated with itself. In our IKM, node B is required to send the accusation in an encrypted form $\{[ID_A, s_B]_{\mathcal{K}_{B,p_i}^{-1}}\}_{k_{B,V}}$ to each $V \in \mathcal{F}(ID_A)$, where $k_{B,V}$ is the shared key with V that can be derived using the method given in [37].

The value of β determines the tradeoff between resilience to D-PKG compromise and communication overhead. The smaller β, the lower the related communication overhead, the less resilient the network is to the compromise of D-PKGs, and vice versa. Specifically, in one extreme case that $\beta = 1$, the communication overhead is the lowest, while the compromise of a D-PKG, say $ID_{X_1} (X_1 \in \Omega)$ which has not been revoked, would allow all the accused whose IDs are mapped by $\mathcal{F}$ to ID_{X_1} to escape revocation. In another

extreme case that $\beta = n$, the network shows perfect resilience to D-PKG compromise, while the related communication overhead is the highest. Therefore, β should be carefully chosen in practice to strike a good balance between these two metrics.

4.3.2 Revocation Generation

Upon receipt of an accusation from B, a D-PKG will simply drop it if the accuser itself has been revoked. Otherwise, the D-PKG saves the accusation after decrypting it and verifying B's signature. To prevent an unrevoked compromised node from falsely accusing legitimate nodes, a node is diagnosed as compromised only when the number of accusations against it reaches the network-wide revocation threshold γ in one key update phase or any other predetermined time window. The choice of γ is application-specific and determines the tradeoff between tolerance of false accusations and compromise detectability: a larger γ means higher-level tolerance of false accusations but lower compromise detectability, and vice versa.

Once the revocation threshold is attained, a key revocation against node A needs to be generated and published. In IKM, to generate a revocation needs the joint efforts of t D-PKGs. For simplicity, we assume that, among $\mathcal{F}(ID_A)$, the D-PKG with the smallest ID acts as the role of revocation leader. We distinguish between two cases. If $\beta \geq t$, each of the t D-PKGs in $\mathcal{F}(ID_A)$ with smallest IDs generates a partial revocation (shown below) sent to the revocation leader. If $\beta < t$, all the D-PKGs in $\mathcal{F}(ID_A)$ should generate a partial revocation and send it to the revocation leader. In addition, the revocation leader sends the accumulated accusations against A to $(t - \beta)$ extra randomly picked D-PKGs, each of which responds with a partial revocation after verifying the accusations.

For ease of presentation, let $\mathcal{A} \subset \Omega$ denote the t D-PKGs participating in revocation generation. Each $V \in \mathcal{A}$ generates a partial revocation $K_{\mathrm{P2}}^V H_1(ID_A)$ accumulated at the revocation leader. The revocation leader can construct a complete revocation from these partial revocations through Lagrange interpolation, which is an application of pairing-based threshold signatures [28], [38]. In particular, a complete revocation is derived as

$$\overline{ID_A} = \sum_{V \in \mathcal{A}} \lambda_V(0) K_{\mathrm{P2}}^V H_1(ID_A) = K_{\mathrm{P2}} H_1(ID_A) (\mathrm{mod}\ q),$$

where $\lambda_V(0)$s are Lagrange coefficients defined in (2). It is possible that one or several members of $\mathcal{A}$ are unrevoked compromised nodes which might send wrongly computed partial revocations. To detect this, the revocation leader checks whether the following equation holds.

$$\hat{e}(\overline{ID_A}, W) = \hat{e}(H_1(ID_A), W_{\mathrm{P2}}). \quad (3)$$

If so, it knows that this revocation is authentic and all other $(t - 1)$ D-PKGs gave correct partial revocations. The equation should hold for a valid revocation because

$$\begin{aligned} \hat{e}(\overline{ID_A}, W) &= \hat{e}(K_{\mathrm{P2}} H_1(ID_A), W) \\ &= \hat{e}(H_1(ID_A), W)^{K_{\mathrm{P2}}} \quad (\hat{e} \text{ is bilinear}) \\ &= \hat{e}(H_1(ID_A), K_{\mathrm{P2}} W) \quad (\hat{e} \text{ is bilinear}) \\ &= \hat{e}(H_1(ID_A), W_{\mathrm{P2}}) \quad (W_{\mathrm{P2}} = K_{\mathrm{P2}} W). \end{aligned}$$

The revocation leader then floods $< ID_A, \overline{ID_A} >$ throughout the network to inform others that A has been compromised.

If (3) does not hold, the revocation leader knows that at least one of the partial revocations is incorrect. Our IKM allows the pinpoint identification of the misbehaving D-PKG(s). To do this, for each received $K_{\mathrm{P2}}^V H_1(ID_A)$, the revocation leader harnesses the preloaded W_{P2}^V to check whether the equation $\hat{e}(K_{\mathrm{P2}}^V H_1(ID_A), W) = \hat{e}(H_1(ID_A), W_{\mathrm{P2}}^V)$ holds. The check should succeed for a valid partial revocation because $W_{\mathrm{P2}}^V = K_{\mathrm{P2}}^V W$ and $\hat{e}$ is bilinear. Otherwise, the revocation leader considers V misbehaving and then issues a signed accusation against it. After identifying all misbehaving D-PKGs in $\mathcal{A}$, the revocation leader solicits the corresponding number of new partial revocations from D-PKGs in $\Omega \backslash \mathcal{A}$, calculates a complete revocation, and verifies it as before. Continuing this process, the revocation leader can form a correct revocation against A, as long as there are at least t well-behaved D-PKGs in Ω.

Our IKM can handle the situation that the revocation leader itself is a compromised node well. If other D-PKGs in $\mathcal{F}(ID_A)$ do not receive a correct revocation against A in a certain time window, they would consider the revocation leader misbehaving and publish signed accusations against it. Then, the D-PKG in $\mathcal{F}(ID_A)$ with the second lowest ID succeeds as the revocation leader and restarts the revocation generation process. We can see that, as long as there is at least one noncompromised D-PKG in $\mathcal{F}(ID_A)$ and there are at least t noncompromised D-PKGs in Ω, a valid accusation against node A can always be generated. In addition, our pinpoint identification mechanism will deter the D-PKGs compromised yet unrevoked from offering invalid partial revocations to avoid being easily caught. Therefore, we expect that a valid revocation will be generated most likely in one round. Also notice that, since whether a D-PKG provides a wrong partial revocation and whether the revocation leader behaves normal are both publicly verifiable, compromised but unrevoked D-PKGs dare not falsely accuse the revocation leader or other D-PKGs in order to avoid being identified.

4.3.3 Revocation Verification

Upon reception of $\overline{ID_A}$, every node verifies it by checking if (3) holds. If so, it should record ID_A in its memory and refuse to interact with node A in future time. In our IKM, each node needs to store the IDs of all the revoked nodes. Assuming that each node ID is of 16 bytes, it costs a node about 4 KB to store 250 IDs of compromised nodes, which is believed to be an acceptable overhead given the increasingly low memory price. Some space-efficient data storage techniques such as Bloom filters [39] may be used to reduce the storage overhead. However, we do not further investigate this issue for lack of space.

In rare cases, the revoked A and/or its conspirators may be the sole connections between parts of the network. Since

they would not further propagate the revocation, there might be some legitimate nodes which cannot receive the revocation. Fortunately, this problem can be greatly mitigated by node mobility. In particular, we require each node to store received revocations for a certain amount of time. When a node meets a new neighbor, it can exchange its stored revocations with that neighbor. If that neighbor offers some unknown revocations, it records the revoked node IDs after verifying those revocations. Since a node can dump stored revocations after a while, the related storage overhead should be affordable.

4.4 Key Update

To withstand cryptanalysis and limit any potential damage from compromised keys, it is a common practice [3], [4], [5], [6], [7], [8] to employ relatively frequent key update. A new key update phase p_{i+1} starts either when phase p_i lasts for more than a predetermined time threshold, or when the number of nodes revoked in p_i has attained a prescribed threshold.

In IKM, each node B can update its public key autonomously as $\mathcal{K}_{B,p_{i+1}} := (H_1(ID_B), H_1(\mathrm{salt}_{i+1}))$, where $\mathrm{salt}_{i+1} = \mathrm{salt}_i + 1$. In other words, B just performs two hash operations, one for generating the phase salt for p_{i+1} and the other for computing the new common public-key element. By contrast, generating the common private-key element $\mathcal{K}^{-1}_{p_{i+1}} = K_{\mathrm{P2}} H_1(\mathrm{salt}_{i+1})$ needs the collective efforts of t D-PKGs in Ω. For simplicity, we assume that $Z \in \Omega$ initiates phase p_{i+1}, though in practice, the D-PKGs should take turns to act as this role to balance their resource usage. Z randomly selects $(t-1)$ other nonrevoked D-PKGs from Ω and sends a request to each of them. Let $\mathcal{A}$ denote these t D-PKGs including Z itself. Each $V \in \mathcal{A}$ uses its secret share to generate a partial common private-key element $K^V_{\mathrm{P2}} H_1(\mathrm{salt}_{i+1})$ accumulated at Z which, in turn, constructs the complete $\mathcal{K}^{-1}_{p_{i+1}}$ using Lagrange interpolation, $\mathcal{K}^{-1}_{p_{i+1}} = \sum_{V \in \mathcal{A}} \lambda_V(0) K^V_{\mathrm{P2}} H_1(\mathrm{salt}_{i+1}) = K_{\mathrm{P2}} H_1(\mathrm{salt}_{i+1})$. Notice that $\mathcal{K}^{-1}_{p_{i+1}}$ is self-authenticating in that every node can check its authenticity by checking if the following equation holds:

$$\hat{e}(\mathcal{K}^{-1}_{p_{i+1}}, W) = \hat{e}(H_1(\mathrm{salt}_{i+1}), W_{\mathrm{P2}}). \tag{4}$$

It is also possible that some D-PKGs in $\mathcal{A}$ might be compromised of yet unrevoked nodes. The method used in revocation generation can be employed as well to deal with this case. As long as there are at least t noncompromised D-PKGs in Ω, a valid $\mathcal{K}_{p^{-1}_{i+1}}$ can always be generated.

To propagate $\mathcal{K}^{-1}_{p_{i+1}}$ securely to all the nonrevoked nodes, we use a variant of the self-healing group key distribution scheme by Liu et al. [40].[5] Let $\Lambda \subset \Psi$ denote the set of nodes revoked until phase p_i (including p_i). D-PKG Z broadcasts the following message:

$$\mathcal{B}_i := \{ID_X\}_{X \in \Lambda} \cup \{\mathcal{U}_j(x) = \xi_j(x) u_j(x) + l_j(x)\}_{j=1,\ldots,i},$$

where $\xi_j(x) = \prod_{X \in \Lambda} (x - ID_X)$. When a nonrevoked node, say B, receives this message, it derives

$$\mathcal{U}_i(ID_B) = \xi_i(ID_B) u_i(ID_B) + l_i(ID_B).$$

Since B knows $v_i(x)$, $l_i(ID_B)$, and $\xi_j(ID_B) \neq 0$ (see Section 4.2.4), it can get $u_i(ID_B) = \frac{\mathcal{U}_i(ID_B) - l_i(ID_B)}{\xi_i(ID_B)}$ and then $[\mathcal{K}^{-1}_{p_i}]^y = v_i(ID_B) + u_i(ID_B)$. Subsequently, node B computes $[\mathcal{K}^{-1}_{p_i}]^x$ using the elliptic curve $\mathrm{E}/\mathbb{F}_{\mathrm{p}}$, thus constructing the complete $\mathcal{K}^{-1}_{p_i}$. In a similar way, all the other nonrevoked nodes can derive $\mathcal{K}^{-1}_{p_i}$ and finish key update. Any revoked node $X \in \Lambda$, however, cannot compute $u_i(ID_X)$ and, thus, $\mathcal{K}^{-1}_{p_i}$ because $\xi_i(ID_X) = 0$. In addition, as long as the number of compromised nodes is no more than t^c, i.e., $|\Lambda \leq t^c|$, the compromised nodes cannot jointly determine $\mathcal{K}^{-1}_{p_i}$ either, as shown in [40].

The above key-update method provides the self-healing capability in the sense that any nonrevoked node can recover $\mathcal{K}^{-1}_{p_j}$ for any phase p_j $(j > i)$, of which it did not receive the key-update broadcast message due to reasons such as mobility, channel errors, and temporary network partitions. Consider node B again as an example. It can get $\mathcal{K}^{-1}_{p_j}$ in the similar way as obtaining $\mathcal{K}^{-1}_{p_i}$. This nice feature, however, is achieved at the cost of increased communication overhead. Therefore, if either this self-healing capability is not required or reliable broadcast can be guaranteed, the broadcast message $\mathcal{B}_i$ can change to $\{ID_X\}_{X \in \Lambda_i} \cup \{\mathcal{U}_i(x) = \xi_i(x) u_i(x) + l_i(x)\}$, where $\xi_i(x) = \prod_{X \in \Lambda} (x - ID_X)$ and $\Lambda_i \subseteq \Lambda$ represents the set of new nodes needed to be revoked in phase p_i. In doing so, the broadcast communication overhead can be reduced.

4.5 Securing D-PKGs against Pinpoint Attacks

Similar to [3], [6], [7], our IKM relies on the validity of the t-limited assumption mentioned in Section 3.3. However, if adversaries have the entire network lifetime to mount attacks, they may compromise or disrupt enough D-PKGs sooner or later. As a well-known countermeasure, Herzberg et al. [35] propose to periodically refresh secret shares without changing the original secret, in such a way that any information learned by adversaries about individual shares becomes obsolete after the shares are refreshed. In addition, they present techniques to periodically and securely recover shares not refreshed properly to withstand D-PKG disruption attacks. Their techniques are either adopted or suggested by [3], [6], [7]. To deal with long-term adversaries, we also suggest to incorporate such proactive secret-sharing techniques in our IKM.

Proactive secret-sharing techniques are valid as long as adversaries are t-limited in each predefined time period. Nearly all previous proposals simply make this assumption without efforts to justify it. In our opinion, without precaution, the t-limited assumption is difficult to hold for MANETs deployed in hostile environments. The reason is that the IDs of the D-PKGs are public knowledge to every node, and adversaries can easily get this information, e.g., by compromising a single node. In common MANET

5. $\mathcal{K}^{-1}_{p_i}$ can be viewed as a group key to be distributed to nonrevoked group members.

routing protocols such as AODV [41] and DSR [42], node IDs are left bare without any protection. The shared wireless medium renders adversaries to perform passive eavesdropping and easily locate the D-PKGs based on their IDs leaked in routing and data packets. As a result, adversaries can launch pinpoint compromise or disruption attacks on the locked D-PKGs. This type of severe pinpoint attacks resulting from the unique characteristics of MANETs are reported in [17], [43]. Obviously, we have to seek efficient ways to thwart such pinpoint attacks to make the t-limited assumption reasonable.

Assume that adversaries have no ways (e.g., traffic analysis) to distinguish between the D-PKGs and non-D-PKG nodes other than from their IDs. We propose to eliminate the pinpoint attacks via our prior work MASK [17], an anonymous on-demand routing protocol for MANETs. Also built upon IBC, MASK can nicely fulfill the routing and packet forwarding tasks without disclosing the real IDs of participating nodes. It is shown to have high routing efficiency comparable to that of classic AODV [41]. For lack of space, we refer to [17] for more details on MASK. Our MASK guarantees that, given a node ID, adversaries cannot ascertain whom and where the corresponding node is. For our purpose, this means that, even given the list of D-PKG IDs, adversaries cannot determine which nodes are the D-PKGs based on passive eavesdropping of node IDs. Therefore, the pinpoint attacks are effectively defeated. Also, note that the same method can be used to eliminate pinpoint attacks on the D-CAs in [3], [6], [7].

4.6 Choosing Secret-Sharing Parameters

Now, we discuss how to select the secret-sharing parameters t, n for a good trade-off between security, and robustness, namely, the resilience to the compromise and disruption of D-PKGs, respectively. For a fixed n, the larger t is, the more secure the network is because adversaries need to compromise more D-PKGs to learn K_{P2}, the less robust the network is in that adversaries need to disrupt fewer D-PKGs to make K_{P2} irrecoverable, and vice versa. To strike a good balance between them, it is often wise to let $t = \lceil \frac{n}{2} \rceil$, as suggested in [14], [15]. The next question is, given the network size N, how we decide the value of n to achieve desired levels of security and robustness.

With our MASK in place, adversaries cannot distinguish between the D-PKGs and common nodes based on passive eavesdropping. What they can only do is to attempt to compromise or disrupt randomly picked nodes with the expectation that those nodes happen to be the D-PKGs. Assume that adversaries can surreptitiously compromise and disrupt up to $N_c \geq t$ and $N_d \geq n - t + 1$ nodes, respectively, in each proactive secret-sharing time period without being detected. We define Pr_c and Pr_d as the probabilities that at least t out of N_c compromised nodes and $(n - t + 1)$ out of N_d disrupted nodes happen to be D-PKGs. In particular,

$$Pr_c = \sum_{i=t}^{\min(n,N_c)} \frac{\binom{n}{i}\binom{N-n}{N_c-i}}{\binom{N}{N_c}} \quad \text{and} \quad Pr_d = \sum_{i=n-t+1}^{\min(n,N_d)} \frac{\binom{n}{i}\binom{N-n}{N_d-i}}{\binom{N}{N_d}},$$

where $t = \lceil \frac{n}{2} \rceil$. In practice, we want both probabilities to be as low as possible. Prior to deployment, the PKG can use the enumerative method to determine the values of t, n for obtaining appropriate values of Pr_c and Pr_d, i.e., meeting desirable levels of security and robustness. For example, when $N = 50$, $N_c = 5$, and $N_d = 7$, we have $Pr_c = 1.19 \times 10^{-4}$ and $Pr_d = 8.53 \times 10^{-5}$ if $n = 10$ and, thus, $t = 5$; when $N = 50$, $N_c = 10$, and $N_d = 14$, we have $Pr_c = 1.8 \times 10^{-5}$ and $Pr_d = 7.88 \times 10^{-4}$ if $n = 20$ and, thus, $t = 10$. Obviously, the success probabilities of such random attacks are pretty low.

During network operation, the network size N may be changing with node join, leave, or failure over time. Accordingly, the parameters t, n and the D-PKG set should be adjusted to maintain desirable levels of security and robustness. This can be easily realized through *verifiable secret redistribution* by Wong et al. [44] to redistribute the PKG's master key K_{P2} from a (t, n) structure to a (t', n') one.

4.7 Security Analysis

Here, we briefly compare the security of our IKM with CKM such as [3], [6] and previous IBC-based schemes [12], [13] (referred to as *o-IKM*). In o-IKM, the PKG only has one master secret K_{P2} jointly shared by n chosen D-PKGs in a (t, n)-threshold fashion. Each node A has a public/private key pair $(H_1(ID_A \| exp), K_{\mathrm{P2}} H_1(ID_A \| exp))$, where exp indicates the key expiration time. To renew its private key before it expires, A needs to individually contact t out of n D-PKGs for partial private keys, based on which to construct a complete one via Langrange interpolation. As usual, our discussion is from the viewpoint of key management instead of cryptographic algorithms themselves.

Since all three approaches are (t, n)-threshold schemes, they have the same level of security as long as the t-limited assumption holds. However, they differ in the worst-case scenario where adversaries manage to compromise at least t distributed CAs (D-CAs for short) in CKM, or t D-PKGs in IKM or o-IKM. In that situation, adversaries are able to construct the CA's private key in CKM, or the PKG's master secret K_{P_2} in IKM or o-IKM. For both CKM and our IKM, adversaries cannot deduce the private key of any non-compromised node, be it a D-CA (or D-PKG) or common node. Therefore, the communication security between noncompromised nodes is still guaranteed. In contrast, the exposure of K_{P_2} in o-IKM would result in the loss of overall system security because it permits adversaries to derive all the private keys of all the compromised or noncompromised nodes ever used since the network formation. This means that adversaries would be able to freely read encrypted messages observed in the past or future, and forge any node's digital signature.

In summary, our IKM is at least as secure as conventional CKM, but outperforms o-IKM in the worst-case scenario.

5 Performance Evaluation

In this section, we compare the proposed IKM with conventional CKM via simulations. As mentioned in Section 2.2, DSA-based CKM solutions have much worse communication efficiency than RSA-based ones under the same security level. Therefore, we focus on comparing IKM with RSA-based CKM, which is implemented mainly based

on [4], [8] with the number of D-CAs set to n instead of N. As discussed before, our IKM is more secure than o-IKM [12], [13] under the same secret-sharing parameters (t, n). In addition, the communication and computation overheads of o-IKM are the same as those of IKM with regard to key revocation, but are much higher in terms of key update because o-IKM requires that each node individually contact t out of n D-PKGs for key update. Since the advantages of our IKM over o-IKM are quite obvious, we do not offer the simulation results of their comparison for lack of space.

5.1 Simulation Setup

The comparison is done within GloMoSim [45], a popular MANET simulator, on a desktop with an Intel P4 2.4GHz processor and 1 GB memory. Although such a powerful machine may not be available in some application scenarios, it should be appropriate for the comparative study of IKM and CKM. To avoid causal implementation errors and guarantee fair comparison, all the cryptographic primitives are built using MIRACL [46], a standard cryptographic library.

For CKM, the underlying CBC is RSA with a 1,024-bit modulus for sufficient security. An RSA public key consists of an ordered pair (s, e) where s is the modulus, and e is the public exponent. A common value for the public exponent is $e = 2^{16} + 1$, which is the value we use for all public exponents. Note that this is in favor of CKM because RSA encryption and signature verification can be made very fast with $e = 2^{16} + 1$ than a random exponent. Therefore, an RSA public key would require 128 bytes for the modulus and 3 bytes for the public exponent, resulting in a total size of 131 bytes. In addition, an RSA signature consists of a single 1,024-bit value. For simplicity, we assume that a node ID is of 16 bytes and that certificate expiration time can be encoded in 2 bytes. An RSA certificate $< ID_A, (n, e), exp, \text{CA's signature} >$ will be a total of 277 bytes in length.

For our IKM, the bilinear map $\hat{e}$ we use is the Tate pairing [29]. q is a 160-bit Solinas prime $2^{159} + 2^{17} + 1$ and p is a 512-bit prime equal to $12qr - 1$ (for some r large enough to make p the correct size). Such choices of q, p deliver a comparable level of security to 1,024-bit RSA [27], [28]. The elliptic curve E we use is $y^2 = x^3 + x$ defined over $\mathbb{F}_p$. The ID-based signature primitive $[M]_{\mathrm{K}^{-1}_{A_{PKG_i}}}$ used is the one outlined in [37], in which a signature consists of one element of $\mathbb{G}_1$ and one element of $\mathbb{Z}^*_q$. Since the former is a point on $\mathrm{E}/\mathbb{F}_p$, only the y-coordinate needs to be transmitted because the x-coordinate can be easily derived using E. Therefore, an ID-based signature is of 84 bytes. This point compression technique is also used in transmitting key revocations and common private-key components, both being elements in $\mathbb{G}_1$. Moreover, the hash function SHA-1 [16] and the symmetric-key encryption primitive RC6 [47] are used wherever applicable.

We simulate a MANET with 50 nodes deployed in a 700×700 m^2 square field.[6] The physical-layer path loss model is the two-ray model. The node transmission range is 250 meters and the channel capacity is 2 Mb/s. The MAC protocol used is the Distributed Coordination Function (DCF) of the IEEE 802.11. For simplicity, the underlying routing protocol is AODV [41] instead of our MASK [13]. Nodes initially are uniformly distributed and node mobility are emulated according to the random waypoint model [42]. We run simulations for constant node speeds of 5, 10, and 15 m/s, with pause time fixed to 5 seconds. In addition, we use 20 CBR connections with random source and destination pairs throughout the simulations. All the data packets are 512 bytes and are sent at a speed of four packets/s.

6. Note that for the simulated network size, it may be feasible to preload each node with all the others' public keys. However, it should be understood that this choice is just for illustration purposes and also to ensure a fair comparison with ARAN [18] which uses the same network size.

TABLE 2
Timings of Primitive Operations

Primitive	Time (ms)
RSA key generation	526.5
RSA encryption/verfication ($e = 2^{16}+1$)	0.26
RSA decryption/signing	5.08
Modular exponentiation ($m^N \bmod N$)	16.89
Map-to-point $H_1(\cdot)$	2.6
Scalar multiplication in $\mathbb{G}_1$	3.3
Modular exponentiation in $\mathbb{G}_2$	2.4
Pairing	11.0
ID-based signing (with pre-computation)	5.7
ID-based signature verification	35.5

5.2 Computational Costs

We present the computational costs of outstanding primitive operations in CKM and IKM in Table 2. As compared to RSA operations, the pairing evaluation is currently a relatively expensive operation, which by far takes the most running time of an IBC algorithm. However, since the pairing is a relatively new technique, we anticipate that its evaluation cost will be much reduced with the rapid advance in cryptography. For example, Barreto et al. [48] recently announced an approach to evaluate the Tate pairing by up to 10 times faster than previous methods, the implementation of which is underway. In addition, the pairing computation can be much accelerated by using dedicated cryptographic hardware. For instance, it is reported in [49] that the Tate pairing can be calculated in about 6 ms on a modern FPGA. Despite its computational inefficiency, we will see below that our IKM still outperforms CKM in almost all aspects because of its certificateless nature.

5.3 Comparison in Key Revocation

Here, we compare IKM with CKM with regard to key revocation. We use 20 CBR sessions as background "noise" to simulate more realistic scenarios. Two sets of secret-sharing parameters (t, n) are simulated: (5, 10) and (10, 20). The revocation process of CKM is implemented as similar to that of our IKM. For simplicity, we set the revocation threshold γ equal to t and each accusation is sent to $\beta = 1$ D-PKG in IKM or D-CA in CKM. In other words, when the number of accusations against one specific node reaches $\gamma = t$ at a D-PKG or D-CA, that D-PKG or D-CA sends the accumulated accusations to other random $(t-1)$ out of $(n-1)$ D-PKGs or D-CAs which, in turn, send back partial revocations after verifying the received accusations. To

TABLE 3
Comparison of Key Revocation Time

	threshold $t = 5$		threshold $t = 10$	
Speed (m/s)	IKM (sec)	CKM (sec)	IKM (sec)	CKM (sec)
5	3.344	3.179	8.563	8.323
10	3.356	3.220	8.577	8.387
15	3.362	3.235	8.586	8.401

avoid possible MAC-layer collisions resulting from returned partial revocations, the revocation leader uses a fixed delay of one second between contacting two different D-PKGs.

Table 3 gives the one-time key revocation time of IKM and CKM for $t = 5$ and 10, respectively. The counted time starts from when a D-PKG or D-CA sends the accumulated accusations to $(t-1)$ peers, until the last node in the network receives and verifies the final complete revocation. All packet transmission and cryptographic processing time has been included. As we can see, although our IKM is slightly inferior to CKM, both can finish a key revocation in a very short duration. This demonstrates the feasibility of real-time public-key revocations in MANETs. We can also observe that, the larger the threshold t is, the more time it takes to finish the revocation process, which is quite intuitive. In addition, node mobility has little impact on the revocation time in that the revocation process only involves the transmission of $2(t-1)$ unicast packets and one network-wide broadcast packet for the final revocation. Such a small amount of traffic can be transmitted before the network topology changes significantly and thus some unicast routes break due to node mobility.

5.4 Comparison in Key Update

In this section, we demonstrate the advantage of our IKM over CKM in terms of key update. Again, 20 CBR sessions are used to emulate normal traffic scenarios. For our IKM, the key update process starts when one D-PKG sends a key update request to other random $(t-1)$ D-PKGs,[7] and finishes when all the network nodes receive and verify the broadcasted common private-key component. For CKM, the key update process lasts from when the first node starts contacting t random D-CAs for key update until the last node finishes its key update through t random D-CAs. To avoid traffic collisions at the D-CAs, a fixed interval of 5 seconds is inserted between two consecutive key updates by two different nodes.[8]

We are interested in two metrics: one-time key update time, including packet transmission time and all cryptographic processing time, and key update overhead in number of packets, which counts all the key requests/replies and the incurred routing control packets. Tables 4 and 5 compare our IKM with CKM with regard to these two metrics for $t = 5$ and 10, respectively. Since a key update process in IKM is similar to a key revocation process, it can be finished in a similarly short period. In contrast, key update in CKM requires a relatively great amount of time and incurs a significantly larger overhead. In addition, the key update time and overhead of both schemes increase with the threshold t, which is of no surprise.

7. The 1-s sending interval is still used.

8. We have tried different interval values and the chosen one can guarantee that almost all the nodes can successively finish their key update within the simulation time.

TABLE 4
Comparison of Key Update $(t = 5)$

	IKM: threshold $t = 5$		CKM: threshold $t = 5$	
Speed (m/s)	Time (sec)	Overhead (packet)	Time (sec)	Overhead (packet)
5	3.173	352	271.088	18556
10	3.182	674	271.965	20846
15	3.189	1328	273.443	22400

TABLE 5
Comparison of Key Update $(t = 10)$

	IKM: threshold $t = 10$		CKM: threshold $t = 10$	
Speed (m/s)	Time (sec)	Overhead (packet)	Time (sec)	Overhead (packet)
5	8.187	662	275.289	37078
10	8.194	1286	276.952	45438
15	8.207	1582	279.978	47501

5.5 Comparison in Secure Routing

A most important use of public-key techniques in MANETs is to secure routing protocols. As noted in [18], most existing secure routing schemes for MANETs rely on the use of public keys and certificates without explicitly discussing how to perform certificate distribution. By contrast, a recent work, called ARAN [18], accounts for certificate distribution. ARAN is an elegant scheme because it is essentially a secured version of classic AODV [41] and thus preserves many nice features of AODV. However, using ID-based public/private keys in place of certificate-based ones can turn ARAN into a much more efficient solution, which is shown as follows.

Due to space limitations, we refer to [18] for detailed descriptions of ARAN. For ease of presentation, we denote the original ARAN by ARAN-CKM and the modification with our IKM by ARAN-IKM. Regarding the overall routing process, ARAN-IKM is the same as ARAN-CKM. Their difference lies in the structures and cryptographic processing of routing control packets, including route discovery/reply/error packets. For example, assuming a source and destination pair of nodes X and Y, a typical route discovery packet (RDP) in ARAN-CKM is of format

$$< \langle \langle RDP, ID_Y, N_X \rangle_{X^{-1}} \rangle_{A^{-1}}, \mathrm{cert}_X, \mathrm{cert}_A > .$$

Here, $\langle m \rangle_{X^{-1}}$ stands for message m with its RSA signature generated under node X's RSA private key X^{-1}; N_X is a monotonically increasing sequence number set by X; cert_X is the RSA certificate of source X (see Section 5.1 for the certificate format); cert_A is the RSA certificate of an intermediate node A attached when A forwards the RDP of X to its own neighbors.[9] Considering the RDP format $< RDP, ID_Y, N_X, ID_X, ID_A >$ in AODV [41], ARAN-CKM adds 778 bytes to the RDP. Suppose the network is in key update phase p_i. In ARAN-IKM, the RDP changes to

$$< [[RDP, ID_Y, N_X]_{\mathcal{K}^{-1}_{X,p_i}}]_{\mathcal{K}^{-1}_{A,p_i}}, ID_X, ID_A > .$$

Therefore, ARAN-IKM increases the RDP in AODV by 168 bytes because of the two ID-based signatures. The

9. Node IDs are included in certificates. Please refer to [18] on how the RDP is processed in a hop-by-hop manner.

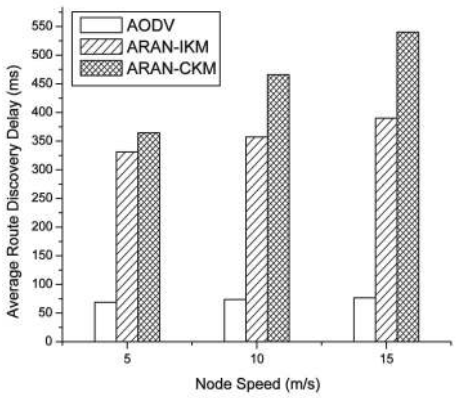


Fig. 1. Average route discovery delay.

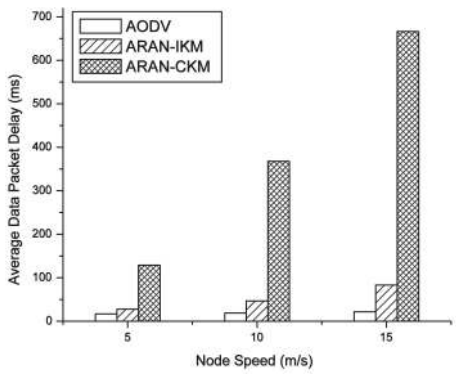


Fig. 2. Average data packet delay.

routing reply and error packets in ARAN-CKM are modified similarly.

We run simulations to compare the routing performance of ARAN-CKM and ARAN-IKM. The results generated with AODV are also provided as the baseline. Again, 20 CBR sessions are used in the simulations and each simulation is executed for 15 simulated minutes. In our simulation results, each data item represents an average of 10 runs with identical traffic models, but with different mobility scenarios.

We use four key performance metrics to evaluate the performance. *Average route discovery delay* measures the average latency from the time of sending a RDP to receiving the first corresponding route reply. *Average data packet delay* measures the average time from the sending of a data packet by a CBR source until its reception at the corresponding CBR destination. This includes all possible delay caused by buffering during route discovery, queuing delay at the interface, retransmission delay at the MAC layer, and propagation and transmission delay at the physical layer. *Packet delivery ratio* (PDR) measures the ratio of the data packets delivered to the destination to those generated by the CBR sources. Finally, *normalized routing load* measures the average amount of routing packet byte transmitted per delivered data packet byte. Each hop-wise transmission of a routing packet byte is counted as one transmission.

The advantages of ARAN-CKM over AODV in the presence of malicious nodes have been demonstrated in [18]. For simplicity, we just compare the performance of AODV, ARAN-CKM, and ARAN-IKM when all the nodes in the network are well-behaved or benign. Note that, no matter whether there are malicious nodes or not, the operations of both ARAN-CKM and ARAN-IKM remain the same. Therefore, as long as we can show that ARAN-IKM outperforms ARAN-CKM in the simulated scenarios, it will also demonstrate better performance than the latter and thus AODV in the face of malicious nodes. In all of our simulation results, AODV always outperforms both ARAN-CKM and ARAN-IKM. This is of no surprise because there are no efforts at all made in AODV to deal with routing attacks. We will focus on discussing the difference between ARAN-CKM and ARAN-IKM.

Fig. 1 compares the average route discovery delay of ARAN-CKM and ARAN-IKM under three mobility scenarios. We can observe that ARAN-IKM always exhibits shorter route discovery delay than ARAN-CKM. The key reason is that routing discovery and reply packets in ARAN-CKM are of much larger sizes than those of ARAN-IKM. As a result, routing packets in ARAN-CKM are more subject to loss due to collisions with other data or routing packets during their transmission. When a source does not receive a route reply packet after sending the RDP for a while, it has to resend the RDP, which worsens the situation. This contributes to the shown advantage of ARAN-IKM over ARAN-CKM. In addition, the performance difference between ARAN-IKM and ARAN-CKM becomes more and more significant with the increase of node mobility. For example, when the node speed is 15 m/s, the route discovery delay of ARAN-IKM is about 390.08 ms, representing a saving of about 28 percent as compared to the 540.32 ms delay of ARAN-CKM. That is because high mobility means that routes will break more frequently, so accordingly route discovery needs to be performed more frequently. Since more routing packets are involved, their probabilities of colliding with other traffic become increasingly higher in ARAN-CKM than in ARAN-IKM.

Fig. 2 plots the average data packet delay versus node speed. As we can see, ARAN-IKM has a significant advantage over ARAN-CKM in all three mobility scenarios. In particular, when the node speed is 5 or 10 or 15 m/s, the data packet delay of ARAN-CKM is about 4.68 or 7.86 or 8.04 times longer than that of ARAN-IKM. This result is partly due to the shorter route discovery delay ARAN-IKM has than ARAN-CKM, which results in shorter delay caused by buffering at the network layer. Another more important reason is that MAC-layer frames in the IEEE 802.11, including RTS/CTS/DATA/ACK, are more subject to collisions with the MAC frames of routing packets in ARAN-CKM than in ARAN-IKM because the former has much larger-sized routing packets. The situation deteriorates with the increase in node mobility and thus the increase in the number of routing packets. As a result, data packets in ARAN-CKM experience much longer queuing and retransmission delay at the MAC layer.

Fig. 3 shows the PDRs of AODV, ARAN-IKM, and ARAN-CKM for three mobility scenarios. In all cases, ARAN-IKM demonstrates performance close to AODV and higher than ARAN-CKM. This mainly results from the fact that a smaller portion of data packets are dropped in ARAN-IKM than in ARAN-CKM due to attainment of the retransmission limit at the MAC layer. The ultimate reason, however, is still because of the larger-sized routing packets in ARAN-CKM. Finally, the normalized routing load of ARAN-IKM and ARAN-CKM are shown in Fig. 4. For node

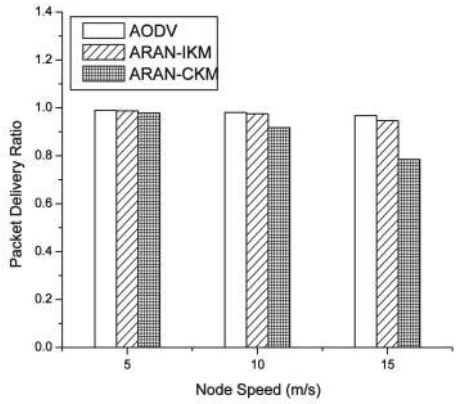


Fig. 3. Packet delivery ratio.

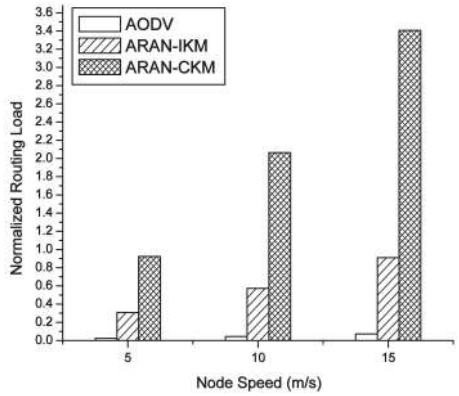


Fig. 4. Average routing load.

speeds of 5 or 10 or 15 m/s, ARAN-CKM has a routing load 3.1 or 3.7 or 4.1 times higher than that of ARAN-IKM for the larger sizes of routing packets.

To summarize, our IKM has significant advantages over conventional CKM in secure routing protocol design, a fundamental component in MANET security.

6 Conclusion

Key management is a fundamental, challenging issue in securing MANETs. This paper presents IKM, a secure, lightweight, scalable ID-based key management scheme for MANETs. As a novel combination of ID-based and threshold cryptography, IKM is a certificateless solution that permits public keys of mobile nodes to be directly derivable from their known network IDs and some other common information. It thus obviates the need for public-key distribution and thus certificates inherent in conventional public-key solutions. Our IKM is characterized by a novel method of constructing ID-based public/private keys, which not only guarantees high-level resilience to node compromise attacks but also facilitates very efficient network-wide key update by a single broadcast message. In addition, we give general guidelines on choosing the secret-sharing parameters for achieving desirable levels of security and robustness. The significant advantages of IKM over conventional certificate-based solutions have been confirmed by extensive simulation results.

Most existing security mechanisms for MANETs thus far involve the heavy use of public-key certificates. In this regard, we believe that the findings of this paper would have much influence on the research paradigm of the whole community and stimulate many other fresh research outcomes. As our future work, we will seek efficient solutions based on IKM to a variety of challenging security issues in MANETs such as intrusion detection and secure routing.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-0626881 and ANI-0093241 (CAREER Award), and the US Office of Naval Research Young Investigator Award N000140210464

References

[1] W. Lou and Y. Fang, "A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions," *Ad Hoc Wireless Networking*, X. Chen, X. Huang, and D.-Z. Du, eds., Kluwer Publisher, Mar. 2003.

[2] B. Neuman and T. Tso, "Kerberos: An Authentication Service for Computer Networks," vol. 32, no. 9, pp. 33-38, Sept. 1994.

[3] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Networks*, vol. 13, no. 6, pp. 24-30, 1999.

[4] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Network Protocols*, Nov. 2001.

[5] M. Narasimha, G. Tsudik, and J.H. Yi, "On the Unitility of Distributed Cryptography in P2P and Manets: The Case of Membership Control," *Proc. IEEE Int'l Conf. Network Protocols*, Nov. 2003.

[6] S. Yi and R. Kravets, "Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks," *Proc. Second Ann. PKI Research Workshop (PKI '03)*, Apr. 2003.

[7] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks," *Proc. IEEE INFOCOM*, Mar. 2004.

[8] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 6, pp. 1049-1063, Dec. 2004.

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

[10] A. Khalili, J. Katz, and W. Arbaugh, "Toward Secure Key Distribution in Truly Ad Hoc Networks," *Proc. IEEE Workshop Security and Assurance in Ad Hoc Networks*, Jan. 2003.

[11] H. Deng, A. Mukherjee, and D. Agrawal, "Threshold and Identity-Based Key Management and Authentication for Wireless Ad Hoc Networks," *Proc. Int'l Conf. Information Technology: Coding and Computing (ITCC '04)*, Apr. 2004.

[12] N. Saxena, G. Tsudik, and J.H. Yi, "Identity-Based Access Control for Ad Hoc Groups," *Proc. Int'l Conf. Information Security and Cryptology*, Dec. 2004.

[13] Y. Zhang, W. Liu, W. Lou, Y. Fang, and Y. Kwon, "AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm*, pp. 3515-3519, May 2005,

[14] A. Shamir, "How to Share a Secret," *Comm. ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[15] Y. Desmedt and Y. Frankel, "Threshold Cryptosystems," *Proc. CRYPTO '89*, pp. 307-315, Aug. 1989.

[16] NIST, "Digital Hash Standard,"Federal Information Processing Standards PUBlication 180-1, Apr. 1995.

[17] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM '05*, pp. 1940-1951, Mar. 2005,

[18] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "Authenticated Routing for Ad Hoc Networks," *IEEE J. Selected Areas Comm.*, vol. 23, no. 3, pp. 598-610, Mar. 2005.

[19] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-Organized Public Key Management for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 2, no. 1, pp. 52-64, Jan.-Mar. 2003.

[20] J.R. Douceur, "The Sybil Attack," *Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '02)*, pp. 251-260, Mar. 2002.

[21] S. Jarecki, N. Saxena, and J.H. Yi, "An Attack on the Proactive RSA Signature Scheme in the URSA Ad Hoc Network Access Control Protocol," *Proc. Second ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '04),* Oct. 2004.

[22] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Comm. ACM,* vol. 21, no. 2, pp. 120-126, Feb. 1978.

[23] NIST, "Digital Signature Standard," Federal Information Processing Standards Publication 186-2, Feb. 2000.

[24] M.G. Gouda and E. Jung, "Certificate Dispersal in Ad-Hoc Networks," *Proc. 24th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '04),* Mar. 2004.

[25] M. Bohio and A. Miri, "Efficient Identity-Based Security Schemes for Ad Hoc Network Routing Protocols," *Elsevier Ad Hoc Networks J.,* vol. 2, no. 3, pp. 309-317, 2004.

[26] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks," *IEEE Trans. Wireless Comm.,* vol. 5, no. 9, pp. 2376-2385, Sept. 2006.

[27] D. Boneh and M. Franklin, "Identify-Based Encryption from the Weil Pairing," *Proc. CRYPTO '01,* pp. 213-229, 2001.

[28] D. Boneh and M. Franklin, "Identify-Based Encryption from the Weil Pairing," *SIAM J. Computing,* vol. 32, no. 3, pp. 586-615, Mar. 2003.

[29] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," *Proc. CRYPTO '02,* pp. 354-368, 2002.

[30] K. Barr and K. Asanovic, "Energy Aware Lossless Data Compression," *Proc. First Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '03),* pp. 231-244, May 2003,

[31] Y. Zhang, W. Lou, and Y. Fang, "SIP: A Secure Incentive Protocol Against Selfishness in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf.,* pp. 1679-1684, Mar. 2004.

[32] R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Adaptive Security for Threshold Cryptosystems," *Proc. CRYPTO '99,* pp. 98-115, Aug. 1999.

[33] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. ACM MobiCom,* Aug. 2000.

[34] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," *Proc. ACM MobiCom '00,* Aug. 2000.

[35] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive Secret Sharing or: How to Cope with Perpetual Leakage," *Proc. CRYPTO '95,* pp. 339-352, 1995.

[36] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols," *Proc. ACM Conf. Computer and Comm. Security,* pp. 62-73, Nov. 1993.

[37] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing Mobile Ad Hoc Networks with Certificateless Public Keys," technical report, Dept. of Electrical and Computer Eng., Univ. of Florida, Gainesville, Apr. 2006.

[38] A. Boldyreva, "Efficient Threshold Signatures, Multisignatures, and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme," *Proc. Sixth Int'l Workshop Theory and Practice in Public Key Cryptography (PKC '03),* Jan. 2003.

[39] B. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," *Comm. ACM,* vol. 13, no. 7, July 1970.

[40] D. Liu, P. Ning, and K. Sun, "Efficient Self-Healing Group Key Distribution with Revocation Capability," *Proc. ACM Conf. Computer and Comm. Security,* Oct. 2003.

[41] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003.

[42] D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks,* vol. 353, pp. 153-181. Kluwer Academic Publishers, 1996.

[43] J. Kong and X. Hong, "ANODR: Anonymous on Demand Routing with Untraceable Routes for Mobile Ad-Hoc Networks," *Proc. ACM MobiHoc '03,* Jun. 2003.

[44] T. Wong, C. Wang, and J. Wing, "Verifiable Secret Redistribution for Archive Systems," *Proc. First Int'l IEEE Security in Storage Workshop,* Dec. 2002.

[45] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large Scale Wireless Networks," *Proc. 12th Workshop Parallel and Distributed Simulations (PADS '98),* pp. 154-161, May 1998.

[46] Shamus Software Ltd., Miracl library, http://indigo.ie/~mscott/, 2005.

[47] R. Rivest, M. Robshaw, R. Sidney, and L. Yin, "The RC6 Block Cipher," v1.1, Aug. 1998, http://www.rsasecurity.com/rsalabs/rc6/.

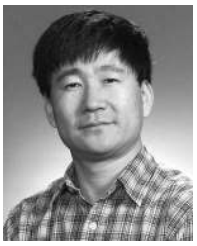
[48] P. Barreto, B. Lynn, and M. Scott, "On the Selection of Pairing-Friendly Groups," *Selected Areas in Cryptography—SAC '03,* pp. 17-25, 2004.

[49] T. Kerins, W. Marnane, E. Popovici, and P. Barreto, "A Hardware Accelerator for Pairing Based Cryptosystems," submittted preprint, 2005, http://paginas.terra.com.br/informatica/paulobarreto.

Yanchao Zhang received the BE degree from Nanjing University of Posts and Telecommunications, China, in 1999, the ME degree from Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, in 2006. He is currently an assistant professor in the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology, Newark. His research interests include network and distributed system security, wireless networking, and mobile computing. He is a member of the IEEE and the ACM.

Wei Liu received the BE and ME degrees in electrical and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001. In August 2005, he received the PhD degree in electrical and computer engineering from the University of Florida. Currently, he is a senior technical member with Scalable Network Technologies. His research interest includes cross-layer design, and communication protocols for mobile ad hoc networks, wireless sensor networks, and cellular networks.

Wenjing Lou received the BE and ME degrees in computer science and engineering from Xi'an Jiaotong University, China, in 1993 and 1996, respectively, the MASc degree from Nanyang Technological University, Singapore, in 1998, and the PhD degree in electrical and computer engineering from the University of Florida in 2003. She is an assistant professor in the Electrical and Computer Engineering Department at Worcester Polytechnic Institute. From Dec. 1997 to July 1999, she worked as a research engineer in Network Technology Research Center, Nanyang Technological University. Her current research interests are in the areas of ad hoc and sensor networks, with emphases on network security and routing issues. She is a member of the IEEE.



Yuguang Fang received the PhD degree in systems engineering from Case Western Reserve University in January 1994 and the PhD degree in electrical engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at the University of Florida in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003, and to a professor in August 2005. He has published more than 150 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the US Office of Naval Research Young Investigator Award in 2002. He has served on many editorial boards of technical journals including the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Mobile Computing*, and *ACM Wireless Networks*. He is a senior member of the IEEE.