Efficient Node Admission for Short-lived Mobile Ad Hoc Networks [*]

Nitesh Saxena, Gene Tsudik
School of Information and Computer Science
University of California, Irvine
{nitesh,gts}@ics.uci.edu

Jeong Hyun Yi[†]
Information Security Technology Group
Samsung Advanced Institute of Technology
jeongy@gmail.com

Abstract

Admission control is an essential and fundamental security service in mobile ad hoc networks (MANETs). It is needed to securely cope with dynamic membership and topology and to bootstrap other important security primitives (such as key management) and services (such as secure routing) without the assistance of any centralized trusted authority. An ideal admission protocol must involve minimal interaction among the MANET nodes, since connectivity can be unstable. Also, since MANETs are often composed of weak or resource-limited devices, admission control must be efficient in terms of computation and communication.

Most previously proposed admission control protocols are prohibitively expensive and require a lot of interaction among MANET nodes in order to securely reach limited consensus regarding admission and cope with potentially powerful adversaries. While the expense may be justified for long-lived group settings, short-lived MANETs can benefit from much less expensive techniques without sacrificing any security. In this paper, we consider short-lived MANETs and present a secure, efficient and a fully non-interactive admission control protocol for such networks. More specifically, our work is focused on novel applications of non-interactive secret sharing techniques based on bi-variate polynomials, but, unlike other results, the associated costs are very low.

1 Introduction

Mobile ad hoc networks (MANETs) have many well-known applications in military settings as well as in emergency and rescue operations. However, lack of infrastructure and lack of centralized control make MANETs inherently insecure, and therefore specialized security services are needed for their deployment. *Admission Control* (or secure node admission) is a fundamental security service in MANETs; it is required to ascertain membership eligibility and to bootstrap other important security services, such as secure routing (e.g., [13, 12]) and secure group communication (e.g., [34, 33]).

Node admission in MANETs cannot be performed centrally. Since, requiring constant presence (availability) of a central fixed entity which is charged with admission control is not realistic for many types of MANETs. First, such an entity is a single point of failure. Second, it represents an attractive and high-payoff target for attacks. Third, topology changes due to mobility and node outages may cause the central entity to be unreachable and thus unable to perform its duties in the parts of a MANET not connected to it. This motivates us to investigate admission control techniques that function in a distributed or decentralized manner. Since our emphasis is on security, the natural technology to consider is threshold cryptography.

The concept of threshold cryptography involves distributing cryptographic primitives (such as decryption or digital signatures) in order to secure them against corruption of a certain number of parties, i.e., a threshold. For example, a (t, n) threshold signature scheme [7] allows, in a group of n parties, to share the ability to digitally sign messages in such a way that any t parties can do so jointly, whereas, no coalition of up to $(t-1)$ parties can. Such a threshold signature scheme is resilient against the so-called *static adversary* who corrupts at most $(t-1)$ parties in the entire lifetime of the system.

More advanced *proactive* cryptographic schemes [11] offer improved resistance against corruptions. Time is divided into *update rounds*, and the proactive scheme offers the same combination of security and robustness even in the presence of so-called *mobile adversaries* [27], whereby a potentially new set of up to $(t-1)$ parties becomes corrupted in each update round. This is done by the *proactive update* procedure which involves parties randomly re-sharing the

[*] This work is supported in part by an award from the Army Research Office (ARO) under contract W911NF0410280, NSF awards under 0331707 and 0331690 (ITR-RESCUE), and a grant from SUN Microsystems.

[†] Corresponding author. This work has been done in part while at UCI.

shared secret at the start of each update round.

Two features of MANETs make decentralized node admission a very challenging problem. *First*, MANET devices often have very weak computational facilities and battery power. *Second*, MANET nodes usually function in an asynchronous (on/off) manner, often becoming temporarily unavailable. Therefore, an ideal admission control protocol must be efficient in terms of both computation and communication[1]. It must also involve minimal (ideally, *none* at all) interaction among the nodes of the network.

A number of admission control techniques have been proposed in recent years [18, 17, 21, 25, 30, 31]. (See the following section for more details on prior work.) Most are based on (t, n) threshold cryptography and allow any set of t-out-of-n nodes (called sponsors) to admit a new node by issuing to it:

(1) a share of a group secret (to be used in future admissions), and

(2) a membership certificate (used for secure communication)

Unfortunately, all previous schemes are far from ideal. They are **heavily interactive** among the sponsors as far as either (1) or (2). Furthermore, they are very computationally expensive in performing (2). This severely limits their practicality.

In this paper, we distinguish between *long-lived* and *short-lived* MANETs. Long-lived MANETs are formed for the long haul and require strong robustness/resilience. They need to be protected against powerful *mobile adversaries* through periodic updates of the secret shares possessed by the nodes [11]. Short-lived MANETs, on the other hand, are ephemeral and need to be resilient against weaker *static adversaries*.

A MANET formed for the duration of a conference program committee meeting (typically, one day) is one example of a short-lived MANET. Another example is a temporary MANET formed by a group of soldiers on a battlefield as they stay in close proximity to each other. A squadron of military aircraft flying in formation also represents a short-lived MANET. Whereas, a MANET formed by a group of college students taking part in a semester-long project course is an example of a *long-lived* MANET. Another example of a long-lived MANET is a flotilla of merchant vessels or military ships sailing together, say, across the Pacific Ocean.

Previous admission control techniques are designed for long-lived MANET settings. We argue that, unlike wired networks, many MANETs fall into the short-lived category and can benefit from much more efficient admission control techniques, without sacrificing any security. In particular, we observe that admission control for short-lived MANETs can be realized by only issuing node-specific secret shares (item (1) above) and obviating the need for expensive membership certificate issuance. Reasons for not needing membership certificates in short-lived MANETs are discussed in detail in Section 3.

Contributions: We construct an efficient and **fully non-interactive** admission control technique and evaluate it in the context of *short-lived* MANETs. In contrast with prior work, our technique does not require any interaction and does not involve any costly reliable broadcast communication among MANET nodes sponsoring admission. We thoroughly analyze our proposal and show that it compares favorably to previous mechanisms (which were designed for long-lived MANETs) modeled for short-lived MANETs.

Although we suggest how the proposed technique can be extended for use in long-lived MANET settings, in this paper we focus only on (more common) short-lived MANETs and the related evaluation.

Organization: The rest of the paper is organized as follows: we first review prior work in Section 2. The generic admission protocol for short-lived MANETs is presented in Section 3, followed by the overview of admission control based on uni-variate polynomial secret sharing (UniAC) in Section 4. We then describe, in Section 5, the proposed admission control mechanism based on bi-variate polynomial secret sharing (BiAC). The detailed performance results, analysis and comparison of BiAC with UniAC are presented in Section 6. Finally, some issues are discussed in Section 7.

2 Related Work

We now review relevant prior work in MANET security. Zhou and Haas [35] first suggested the use of threshold cryptography to secure mobile ad hoc networks. Their idea was to distribute the trust among the nodes of the network such that no less than a certain threshold of nodes are trusted. They proposed a distributed certification authority (CA) which issues certificates (using some threshold signature [7] protocol) to nodes joining the network. These certificates enable nodes to communicate with each other in a confidential and authenticated manner. This work also led to the development of COCA [36], an on-line certification authority for wired networks. Although quite attractive, this idea is not directly applicable for the purposes of admission control in MANETs. The proposed approach is hierarchical in the sense that only select nodes can serve as parts of the certification authority, i.e., take part in admission decisions. Moreover, contacting distributed CA nodes in a multi-hop and ever-changing MANET might not always be possible.

[1] Communication is directly related to the consumption of battery power in MANET devices [1].

Kong, et al. considered the same problem in series of papers [18, 17, 21, 20] and proposed a set of protocols for providing ubiquitous and robust admission control for MANETs. They adapted the model of Zhou and Haas so that any node can participate in admission control decisions, thus maintaining the true "peer" nature of a MANET and providing increased availability. The security of their admission mechanism relies upon a specific variant of the proactive threshold RSA signature scheme. Unfortunately, this scheme is neither robust [25] (i.e., it can not tolerate malicious nodes) nor secure [15].

Recently, Narasimha, et al. [25] and Saxena, et al. [31] proposed similar admission control protocols based on threshold DSA [9] and threshold BLS [4] signatures, respectively. While provably secure, both solutions are quite inefficient.

As pointed out in the previous section, all of the above techniques are proposed for long-lived MANETs. They require admitting nodes to interact in order to issue a new node its secret share and/or a membership certificate. Both heavy interaction and costly cryptographic computation make these techniques overly expensive for most MANET applications.

The admission control technique developed in this paper is designed for short-lived MANETs and is completely non-interactive. It uses secret sharing based on so-called bi-variate polynomials which have been employed for related purposes in the literature [2, 24, 3]. In particular, [19] presents a key pre-distribution scheme for sensor networks using bi-variate polynomials [3] *in the presence of a centralized authority*. The protocol we propose is fully distributed and allows nodes in a MANET to readily and efficiently share pairwise secret keys without any centralized support.

3 Generic Admission Control Protocol

We claimed earlier (in Section 1) that admission control for short-lived MANETs can be realized by *only* issuing node-specific secret shares. Whereas, for long-lived MANETs, it is also necessary to issue individual node membership certificates. We now discuss the reasoning behind this claim.

In both long-lived and short-lived MANETs, threshold secret sharing is employed to share the group secret using a polynomial of degree $(t-1)$, and every node receives a share (called a secret share) of the group secret.

In long-lived MANETs, nodes need to proactively update [11] their secret shares to defend against mobile adversaries. This involves updating all coefficients of the secret sharing polynomial, except the constant term (which is the actual group secret), and broadcasting a commitment to the polynomial[2]. However, due to the dynamic and asynchronous nature of the MANETs, it is not always possible for each node to receive updated commitment values. Therefore, the only way to bind the commitment to a node's secret share with the group secret (commitment to which remains constant throughout the lifetime and becomes part of the group public key) is by issuing membership certificates to the nodes signed using the group secret. These certificates are then used for authentication and pairwise key establishment purposes.

In short-lived MANETs, since there is no need for proactive updates, the polynomial used for sharing the group secret *remains constant* throughout the lifetime of the MANET and the commitment to this polynomial becomes a part of the group public key. The commitment to each node's secret share is derivable from (and thus automatically bound to) the group public key. Therefore, **node-specific membership certificates are not needed** in short-lived MANETs. The nodes can use their secret shares (and/or the group public key) for the purpose of secure communication with each other.

We define an admission control mechanism for short-lived MANETs as a set of three components:

1. *Initialization:* The group is initialized by either a trusted dealer or a set of founding members. The dealer or founding members initialize the group by choosing a group secret key, and computing and publishing the corresponding public parameters in the group certificate [16]. The group secret is shared among the founding member(s) in such a way that any set of t members can reconstruct it. The share of the group secret possessed by each member is referred to as its *secret share*.

2. *Admission:* A prospective member M_{new} who wishes to join the group must be issued its secret share by current member nodes. M_{new} initiates the admission protocol by sending a `JOIN_REQ` message to the network. A member node, that receives this `JOIN_REQ` message and approves the admission of M_{new}, replies, over a secure channel, with a partial secret share (derived from its secret share) for M_{new}. Once M_{new} receives partial secret shares from at least t different nodes, it uses them to compute its secret share.

   During the above process, a malicious node can easily preclude a prospective node from being admitted by inserting incorrect partial secret shares, i.e., a denial-of-service (DoS) attack. To prevent this, a prospective node must be able to verify the validity of its reconstructed secret share before using them. This feature is

[2] A commitment to a polynomial is a commitment to each of its coefficients.

called *verifiability* in the rest of the paper. Also, when the node detects that its secret share is invalid, it must be able to trace the bogus shares and the malicious node(s) in the MANET. This functionality is provided by the so-called *traceability* feature. Note that *verifiability* is always required, whereas, *traceability* is only necessary when a node detects (via verifiability) that its reconstructed secrets are not valid.

3. *Pairwise Key Establishment:* Each node can use its secret share and/or the public parameters to compute pairwise keys with any other node. This allows nodes to securely communicate with each other.

4 UniAC: Admission Control using Univariate Polynomial Secret Sharing

In this section, we briefly describe previously proposed admission control methods [18, 17, 21, 25, 30, 31] adapted for short-lived MANETs. These methods are based on univariate polynomial secret sharing; we refer to them collectively as: *Uni*Variate *A*dmission *C*ontrol (UniAC). UniAC involves the following steps (for protocol message flows, see Figure 2).

1. *Initialization:* The system can be initialized by a trusted dealer TD or a set of founding nodes. As in Shamir's secret sharing [32] based on a univariate polynomials, the TD (or founding members) choose(s) a large prime q, and select(s) a polynomial

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \pmod q$$

such that $f(0) = S$, where a_i-s are the coefficients of the polynomial, q is a large prime, and S is the group secret. The TD computes each node's secret share ss_i such that $ss_i = f(id_i) \pmod q$, and securely transfers ss_i to node M_i. [Any group of t members who have their shares can recover the secret using Lagrange interpolation: $f(0) = \sum_{i=1}^{t} ss_i \, \lambda_i(0) \pmod q$, where $\lambda_i(x) = \prod_{j=1, j\neq i}^{t} \frac{x - id_j}{id_i - id_j} \pmod q$.]

TD also publishes a commitment to the polynomial as in *Verifiable Secret Sharing* (VSS) [8]. VSS setup involves a large prime p such that q divides $p-1$ and a generator g which is an element of $\mathbb{Z}_p^*$ of order q. TD computes W_i, called the **witness**, such that $W_i = g^{a_i} \pmod p$ for all $i \in [0, t-1]$, and publishes these W_i-s in the group certificate.

2. *Admission:* During the admission protocol, a new node M_{new} is given the partial secret share as $pss_j(new) = ss_j \lambda_j(id_{new})$ by a sponsoring node M_j. Upon receiving these partial share values from t admitting nodes, M_{new} obtains its secret share ss_{new} by simply adding them. It then performs the verifiability checking and, if needed, the traceability procedure. (See [6] for details regarding the actual computations involved in these procedures.)

Note that, in order to compute Lagrange coefficients $\lambda_j(id_{new})$, t sponsoring nodes need to be aware of each other's id-s. Also, since $\lambda_j(id_{new})$-s are publicly known, M_{new} can derive ss_j from $pss_j(new)$. This is prevented using the *random shuffling* technique proposed in [11] by adding extra random value R_{ij} to each share. These R_{ij}-s are securely shared between sponsors M_i and M_j and sum up to zero by construction. (See [11] for details.) This process is also illustrated in Figure 1(a).

We note that, due to the random shuffling procedure, this admission protocol becomes heavily interactive among the t sponsoring nodes – it requires $O(t^2)$ point-to-point messages as well as extremely expensive $O(t)$ reliable broadcast messages [5]. All this makes it impractical for most MANET settings.

3. *Pairwise Key Establishment:* Any pair of nodes M_i and M_j can establish shared keys using their respective secret shares ss_i, ss_j and public VSS information. M_i computes:

$$g^{ss_j} = \prod_{k=0}^{t-1} (W_k)^{{id_j}^k} \pmod p$$

from the public witness values, and exponentiates it with its share ss_i to get a key $K_{ij} = (g^{ss_j})^{ss_i} \pmod p$. Similarly, M_j computes:

$$g^{ss_i} = \prod_{k=0}^{t-1} (W_k)^{{id_i}^k} \pmod p$$

and exponentiates it with its share ss_j to get a key $K_{ji} = (g^{ss_i})^{ss_j} \pmod p$. Since, $K_{ij} = K_{ji}$, M_i and M_j now have a shared secret key.

This key establishment procedure remains secure under the computational Diffie-Hellman (CDH) assumption[3]. In other words, an adversary who corrupts at most $(t-1)$ nodes can not compute a shared key between any pair of honest nodes, as long as the CDH assumption holds.

5 BiAC: Non-interactive Admission Control

We now describe a new admission technique for short-lived MANETs. It is based on secret sharing using bi-

[3] CDH assumption: In a cyclic group generated by $g \in Z_p^*$ of order q, for $a, b \in \mathbb{Z}_q^*$, given (g, g^a, g^b), it is hard to compute g^{ab}.

variate polynomials and is fully non-interactive. We call the protocol *Bi*Variate *A*dmission *C*ontrol (BiAC).

5.1 Overview

As shown in Figure 1(b), we avoid interaction among sponsoring nodes by using a *bi-variate* polynomial $f(x, y)$. Bi-variate polynomials have been previously used for related purposes [2, 24, 3].

To distribute shares among n nodes, a trusted dealer chooses a large prime q, and selects a random symmetric bi-variate polynomial $f(x, y) = \sum_{\alpha=0}^{t-1} \sum_{\beta=0}^{t-1} f_{\alpha\beta} x^\alpha y^\beta \pmod{q}$ such that $f(0, 0) = S$, where the constants $f_{\alpha\beta}$-s are the coefficients of the polynomial and S is the group secret. Since the polynomial is symmetric, $f_{\alpha\beta} = f_{\beta\alpha}$ for each α, β and $f(x, y) = f(y, x)$. For each node M_i, the dealer computes a uni-variate polynomial, called a *share-polynomial*, $b_i(x)$ of degree $(t - 1)$ such that $b_i(x) = f(x, id_i) \pmod{q}$, and securely transfers $b_i(x)$ to each node M_i. Note that, after initializing at least t nodes, the dealer is no longer needed.

In order to admit a new node M_{new}, the current member nodes must issue it a *share-polynomial* $b_{new}(x)$ in a distributed manner. This can be achieved if at least t member nodes provide M_{new} with partial shares $b_j(id_{new})$ such that $b_j(id_{new}) = f(id_{new}, id_j)$ for some $j \in [1, n]$. M_{new} can then use the standard Gaussian elimination procedure [29] to compute $f(id_{new}, x)$, which is the same as $f(x, id_{new})$ (since the polynomial $f(x, y)$ is symmetric) and thus obtain its share-polynomial $b_{new}(x) = f(x, id_{new})$ from t partial shares $b_j(id_{new})$.

Unlike protocols based on sharing of uni-variate polynomials, this scheme *does not* require any interaction among the admitting member nodes.

5.2 Initialization

In *BiAC*, the MANET can be initialized by one node (centralized initialization) or a set of nodes (distributed initialization).

Centralized Initialization: the trusted dealer TD computes a two-dimensional sharing of the secret by choosing a random bi-variate polynomial:

$$f(x, y) = \sum_{\alpha=0}^{t-1} \sum_{\beta=0}^{t-1} f_{\alpha\beta} x^\alpha y^\beta \pmod{q}$$

such that $f(0, 0) = S$. TD computes $W_{\alpha\beta}$, called a **witnesses**, such that $W_{\alpha\beta} = g^{f_{\alpha\beta}} \pmod{p}$ for all $\alpha, \beta \in [0, t - 1]$, and publishes these $W_{\alpha\beta}$-s as part of the group certificate. Once TD computes the witness matrix, it sends each node M_i ($i \in [1, n]$) a distinct *share-polynomial:* $b_i(x) = f(x, id_i)$. TD's presence is needed only during this initialization phase in order to bootstrap the system.

Distributed Initialization: alternatively, the network can be initialized by a set of t or more founding nodes. These nodes agree on a random bi-variate polynomial $f(x, y)$ using so-called *Joint Secret Sharing (JSS)* technique [10].

5.3 Admission Process

In order to join the network, a M_{new} must collect at least t partial shares of the polynomial from t current nodes. Figure 3 shows the protocol message flow for the node admission process[4].

Table 1. Notation

SL_{new}	sponsors list for M_{new}
PK_i	temporary public key of M_i
$S_i(m)$	M_i's signature on message m
K_{ij}	pairwise key between M_i and M_j
$E_{K_{ij}}$	encryption with K_{ij}

$M_{new} \rightarrow M_i$:	$id_{new}, PK_{new}, S_{new}(id_{new}, PK_{new})$	(1)
$M_{new} \leftarrow M_i$:	$id_i, PK_i, S_i(id_i, PK_i)$	(2)
$M_{new} \rightarrow M_j$:	$SL_{new}, S_{new}(SL_{new})$	(3)
$M_i \longleftrightarrow M_j$:	*Random Shuffling*	(4)
$M_{new} \leftarrow M_j$:	$E_{K_{newj}}\{pss_j(new)\}$	(5)

Figure 2. UniAC **Admission Protocol**

$M_{new} \rightarrow M_i$:	$id_{new}, PK_{new}, S_{new}(id_{new}, PK_{new})$	(1)
$M_{new} \leftarrow M_i$:	$id_i, PK_i, E_{K_{newi}}\{b_i(id_{new})\},$ $S_i(id_i, PK_i, E_{K_{newi}}\{b_i(id_{new})\})$	(2)

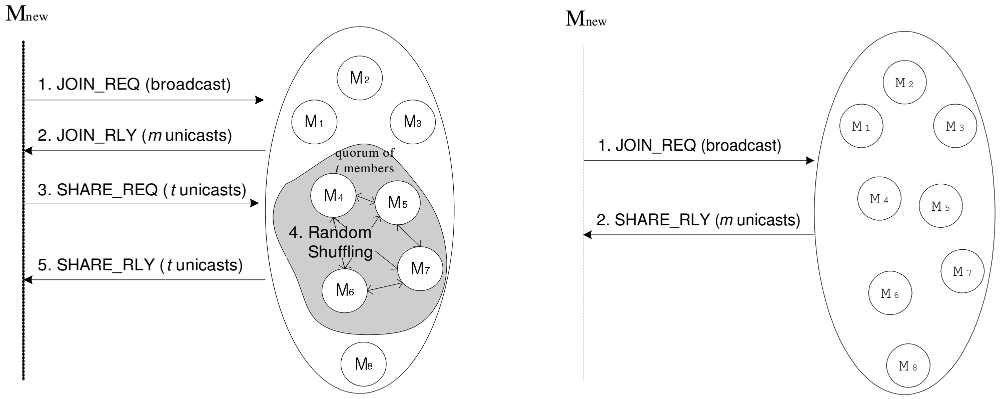
Figure 3. BiAC **Admission Protocol**

The protocol involves following steps (the notation used in this section is summarized in Table 1):

1. M_{new} broadcasts a signed `JOIN_REQ` message which contains its identity id_{new} and its temporary public key PK_{new}. The details about how id_{new} is generated and verified are discussed in Section 7.

2. After verifying the signature on the `JOIN_REQ` message, each receiving node (M_i) willing to admit M_{new}, computes a *partial share* $b_i(id_{new})$ using its own *share-polynomial* such that $b_i(id_{new}) = f(id_{new}, id_i)$. Each sponsor M_i then replies to M_{new} with a `SHARE_REP` message. Each message is signed by the sender and contains encrypted $b_i(id_{new})$ along with the respective values of id_i and PK_i.

[4] In order to secure the protocol against common attacks such as *replay*, *impersonation*, and *interleaving* attacks [22], we note that it is necessary to include additional information such as timestamps, nonces, and identity information of the sender as well as the receiver. However, in order to keep our description simple, we omit these values.



(a) UniAC

(b) BiAC

Figure 1. Comparison of UniAC and BiAC

Note that, in order to compute their partial shares above, sponsors do not need to be aware of each other and thus no interaction is needed. This is in contrast with UniAC scheme, where each sponsor needs to be aware of all other sponsors in order to compute the Lagrange coefficient in partial share issuance, and where interaction is required for random shuffling.

3. Upon receiving m $(\geq \ t)$ SHARE_REP messages, M_{new} selects **any** t of them and interpolates its own share-polynomial $b_{new}(x)$ using standard Gaussian elimination. Let us denote the share-polynomial $b_{new}(x)$ reconstructed by M_{new} as $\sum_{\alpha=0}^{t-1} A_\alpha x^\alpha$. Since $b_i(id_{new}) = b_{new}(id_i)$ (due to symmetry), the problem to interpolate $b_{new}(x)$ using t $b_i(id_{new})$-s is equivalent to the problem to solve the matrix A such that $XA = B$ as follows:

$$\begin{bmatrix} (id_1)^0 & (id_1)^1 & \cdots & (id_1)^{t-1} \\ (id_2)^0 & (id_2)^1 & \cdots & (id_2)^{t-1} \\ & & \vdots & \\ (id_t)^0 & (id_t)^1 & \cdots & (id_t)^{t-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{t-1} \end{bmatrix} = \begin{bmatrix} b_{new}(id_1) \\ b_{new}(id_2) \\ \vdots \\ b_{new}(id_t) \end{bmatrix}$$

The above system of linear equations yields a unique solution since the id_i values are distinct and the matrix $X = [x_{ij}]$, where $x_{ij} = (id_i)^{j-1}$ for all $i, j \in [0, t]$, is invertible.

Verifiability: In order to verify the acquired share-polynomial $\sum_{\alpha=0}^{t-1} A_\alpha x^\alpha$, M_{new} must perform the verifiability procedure. In order to be a valid share-polynomial, A_α must be equal to $\sum_{\beta=0}^{t-1} f_{\alpha\beta}(id_{new})^\beta$, for all $\alpha \in [0, t-1]$. Using the public witness values (from the group certificate) $W_{\alpha\beta} = g^{f_{\alpha\beta}} \pmod p$, the polynomial can be verified as follows:

$$g^{A_\alpha} = \prod_{\beta=0}^{t-1} (W_{\alpha\beta})^{(id_{new})^\beta} \pmod p$$

for all $\alpha \in [0, t-1]$.

Note that the right-hand side in the above equation can be pre-computed by M_{new} prior to starting the admission process.

Traceability: If verification fails, M_{new} can trace the faulty share provider(s) by performing the traceability procedure. This involves verifying the validity of each partial share $b_i(id_{new}) = f(id_{new}, id_i)$, that M_{new} received. This can be achieved by checking the following equation for each i:

$$g^{b_i(id_{new})} = \prod_{\alpha=0}^{t-1} \prod_{\beta=0}^{t-1} (W_{\alpha\beta})^{(id_{new})^\alpha (id_i)^\beta} \pmod p$$

Note that $\prod_{\alpha=0}^{t-1} (W_{\alpha\beta})^{(id_{new})^\alpha}$ in the above equation can be pre-computed since $W_{\alpha\beta}$-s and id_{new} are known to M_{new} in advance.

5.4 Pairwise Key Establishment

Once every node has its share-polynomial, pairwise key establishment is the same as in [3] and [19]. Any pair of nodes M_i and M_j can establish shared keys as follows: M_i uses its share-polynomial $f(x, id_i)$ to compute

$$K_{ij} = f(id_j, id_i) \pmod q$$

and M_j its share-polynomial $f(x, id_j)$ to compute

$$K_{ji} = f(id_i, id_j) \pmod q.$$

Since $f(x, y)$ is a symmetric polynomial, $K_{ij} = K_{ji}$. Thus, M_i and M_j now have a shared key that can be used for secure communication.

Unlike the pairwise key establishment in UniAC (security of which is based on the CDH assumption) as described in Section 4, the security of above procedure is unconditional, i.e., not based on any assumption. Refer to [3] for details regarding the security arguments of this pairwise key establishment.

Table 2. Feature Comparison

Key Features	UniAC	BiAC
Security Assumption (for Admission)	DL	DL
Security Assumption (for Key Comp.)	CDH	Unconditional
Decentralized Admission	Yes	Yes
DoS Resistance	Yes	Yes
Interaction among Sponsors Required	Yes	No
Random Shuffling Required	Yes	No
Reliable Broadcast Required	Yes	No

Table 3. Computation Complexity

Category			UniAC	BiAC
Admission	M_i's view	$\mathcal{M}$	$O(t)$	$O(t)$
		$\mathcal{E}$	$O(t)$	0
	M_{new}'s view	$\mathcal{M}$	$O(t)$	$O(t^3)$
		$\mathcal{E}$	1	$O(t)$
Traceability		$\mathcal{M}$	$O(t^2)$	$O(t^2)$
		$\mathcal{E}$	$O(t^2)$	$O(t^2)$
Pairwise Key Establishment		$\mathcal{M}$	$O(t)$	$O(t)$
		$\mathcal{E}$	$O(t)$	0

$\mathcal{M}$: modular multiplication $\mathcal{E}$: modular exponentiation

Table 4. Communication Complexity

Category		UniAC	BiAC
Rounds	broadcast	1	1
	unicast	$O(t^2)$	$O(t)$
	reliable broadcast	$O(t)$	0
Bandwidth	$\log q$-bit	$O(t^2)$	$O(t)$
	$\log p$-bit	$O(t)$	$O(t)$

6 Performance Analysis

In this section we discuss the implementation of UniAC and BiAC and compare them in terms of node admission, traceability and pair-wise key establishment costs. We also summarize and compare some salient features in Table 2. As expected, BiAC significantly outperforms UniAC in our overall evaluation.

6.1 Complexity Analysis and Comparison

We summarize computation and communication complexities[5] in Tables 3 and 4, respectively, where $n \geq m \geq t$. More specifically, BiAC requires each sponsoring node M_i to perform $O(t)$ modular multiplications and the joining node M_{new} to perform $O(t^3)$ modular multiplications for Gaussian elimination and $O(t)$ exponentiations for verifiability. On the other hand, UniAC requires each M_i to perform $O(t)$ multiplications, and M_{new} to perform $O(t)$ multiplications plus 1 exponentiation for verifiability. For traceability, both the schemes require $O(t^2)$ multiplications and $O(t^2)$ exponentiations with pre-computation. BiAC is significantly more efficient than UniAC for computing pairwise keys, since the former requires only $O(t)$ multiplications, while the latter needs $O(t)$ exponentiations as well as $O(t)$ multiplications. Note that, pairwise key establishment is a very frequent operation in a MANET, thus, its efficiency is extremely important.

As far as overall communication costs[6], BiAC consumes $O(t \log q)$ and $O(t \log p)$ bits, while bandwidth consumption in UniAC is $O(t^2 \log q)$ plus $O(t \log p)$ bits due to the interactive random shuffling procedure.

6.2 Experimental Setups

UniAC and BiAC protocols have been implemented over the popular OpenSSL library [26]. The source is written in C in Linux and consists of about $10,000$ lines of code for each protocol. The code is available at [28].

We used five laptops in our experimental set-up: four with Pentium-3 800MHz CPU-s and 256MB memory and one with Mobile Pentium 1.8 GHz CPU and 512MB memory. Each laptop ran Linux 2.4 and was equipped with a $802.11b$ interface configured for ad-hoc mode. Specifically, for measuring the admission cost, four laptops with the same computing power were used as current member nodes and the high-end laptop was used as the joining/new node. Traceability and pairwise key computation experiments were also performed with this high-end laptop. In our experiments, each node (except the joining node) was emulated by a daemon and each machine was running up to three daemons. The measurements were performed with different threshold values t. The size of the parameter q was set to be 160-bits and p 1024-bits.

To measure consumption of battery power, we performed the following experiment: the test machine was an iPAQ (model H5555) running Linux (Familiar-0.7.2). The CPU on iPAQ is a 400 MHz Intel XScale with 48MB of flash memory and 128MB of SDRAM. In order to obtain accurate power measurements, we removed the battery from the iPAQ during the experiment and placed a resistor in series

[5] The costs required for protecting each protocol message are not taken into account since these costs vary with the specific signature scheme.

[6] We assume that the identity and the public key are $\log q$ bits long and $\log p$ bits long, respectively.

with the power supply. We used a National Instruments PCI DAQ (Data AcQuisition) board to sample the voltage drops across the resistor to calculate current at 1000 samples per second.

6.3 Experimental Results

We compare our experiment results in terms of admission, energy consumption for admission, traceability, and pairwise key computation.

6.3.1 Admission Results

To evaluate admission cost, we measured total processing time between sending of `JOIN_REQ` by the prospective member and receiving (plus verification) of acquired secret shares. Our measurements include the average computation time of the basic operations (such as modular multiplications, exponentiations etc.) as well as communication costs, such as packet en/decoding time, network delay, and so on.

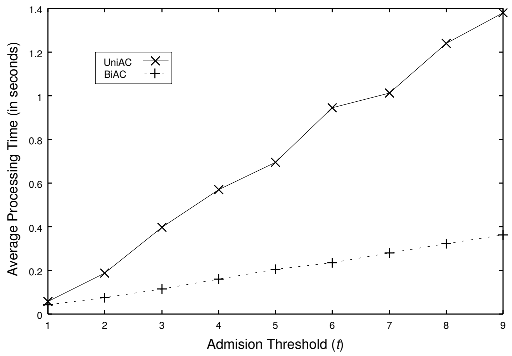


Figure 4. Admission Costs

As observed from Figure 4, the admission (join) cost with BiAC is much lower than that with UniAC. The difference is even higher for higher threshold values. The reason is quite intuitive: not only is BiAC computationally cheaper than UniAC, but it also requires less communication.

Energy consumption results for admission operation are plotted in Figure 5. This experiment is quite tricky to measure fairly. Energy consumption is directly proportional to processing time. It is meaningless to measure energy consumption based on computation time. However, it is well known that, in many small devices such as low-end MANET nodes or sensors, sending a single bit is roughly equivalent to performing 1,000 32-bit computations in terms of batter power consumption [1]. Therefore, we measured the power consumption in terms of communication bandwidth required by each admission protocol. In more detail, we sent some bulk data (e.g., 100 Mbytes) from a single iPAQ PDA, measured the power consumed while sending out this data, and then computed the average power consumption per bit. After that, we calculated the power consumption of each admission protocol by multiplying this measurement result by the bit length of the transmitted data. These results in Figure 5 clearly illustrate that BiAC is much more energy-efficient than UniAC.

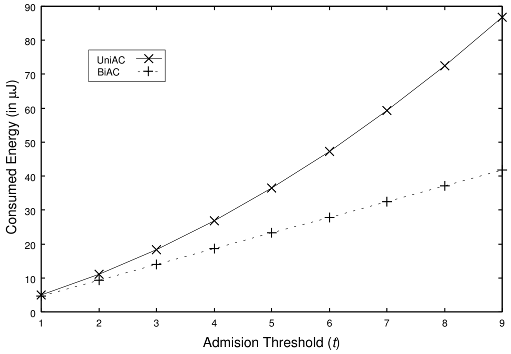


Figure 5. Energy Consumption with Communication for Admission (iPAQ-H5555: XScale 400 MHz, 128MB)

6.3.2 Traceability Results

Figure 6 displays traceability costs for the two approaches. Even in the worst case, BiAC is as good as UniAC for performing the (very infrequent) operation of tracing malicious nodes.

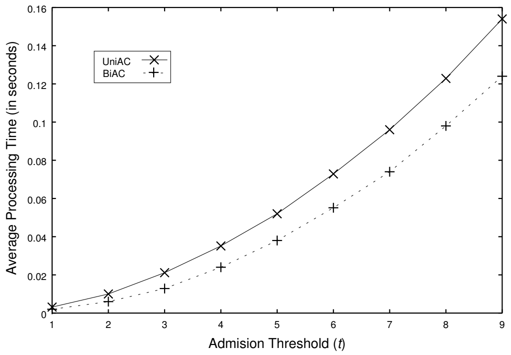


Figure 6. Traceability Costs

6.3.3 Pairwise Key Establishment Results

Figure 7 shows that BiAC is significantly more efficient than UniAC for computing pairwise keys. The achieved gains range approximately from 115 ($t = 1$) to 412 ($t = 9$); in other words, BiAC is 115 to 412 times faster than UniAC when establishing a shared secret key. This result was actually expected because in BiAC the pairwise key computation requires only $O(t)$ multiplications where the modular size is 160 bits. In contrast, UniAC requires $O(t)$ exponentiations with a modular size of 1024 bits as well as $O(t)$ multiplications with 160-bit modulus.

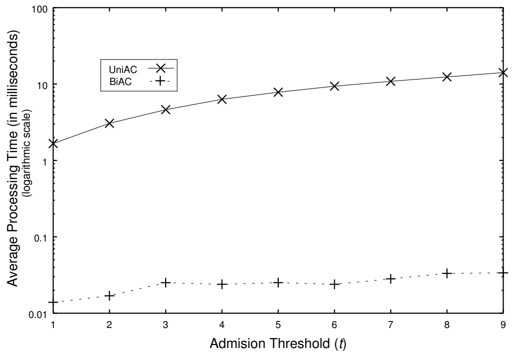


Figure 7. Pairwise Key Establishment Costs

7 Discussion

In this section, we discuss the security of the proposed BiAC scheme and several related issues.

Security. Security of BiAC is based on the discrete logarithm (DL) assumption[7] as long as the adversary is not allowed to corrupt more than $(t-1)$ ($< n/2$, where n is the total number of nodes) nodes in the network. Due to lack of space, we only present a sketch of this argument. Basically, as in the Feldman's VSS [8], we use the idea of *simulated adversarial view* to show that an adversary who corrupts at most $(t-1)$ nodes learns nothing extra (other than the witness $g^S \pmod p$) about the secret S during the initialization and admission procedures of the scheme. This is achieved by generating a simulator, which on input $g^S \pmod p$, produces public information and the private information to the adversary which is statistically indistinguishable from the one produced in the actual run of these procedures. For the security arguments of BiAC pairwise key establishment, we refer the reader to [3].

Identifier Configuration. In the UniAC and BiAC schemes, the identifier id_i of each node M_i must be *unique* and *verifiable*. Otherwise, a malicious node could use the identifier of some other node and obtain its secret from the member nodes during the admission process. For unique and unforgeable ID assignment, we use a solution based on *Crypto-Based ID (CBID)* [23]: The id_i is chosen by the node itself from an ephemeral public key (PK_i) such that $id_i = H(PK_i)$, where $H(\cdot)$ is a one-way collision-resistant hash function. Refer to [23] for details.

Secure Channel Establishment. In the proposed admission protocols, the channels between the node requesting admission and each of the member nodes must be authenticated and encrypted. Establishing an authenticated and encrypted channel usually requires the use of certificates, which bind identities to public keys, and an access to a PKI. However, PKI is not always available in MANET environments. Fortunately in our case, what is really needed is a way to bind an identifier to a public key. This binding is actually provided by *CBID*, described previously. As a result, certificates and PKI are not required.

Using Secret Shares as Private Keys. We make an interesting observation that, for short-lived MANETs, each node M_i can use its share ss_i of the group secret S as its individual private key, the corresponding public key y_i being the commitment to the secret key, such that $y_i = g^{ss_i} \pmod p$. (Recall that this public key can be computed using the public witnesses of the secret sharing polynomial.) The secret key and public key-pairs (ss_i, y_i) can then be used for the purposes of signing and encryption, respectively, to communicate securely with nodes outside (and also inside) the network. Such use of secret shares as private keys [14] is different from the corresponding usage in standard public key cryptosystem because the secret shares are not *independent* (they are points on a curve defined by a polynomial) and at most $(t-1)$ of these values are known to the adversary.

Extension to long-lived MANETs. BiAC can be easily extended for long-lived MANETs. Recall from the Section 1 that, for long-lived MANETs, each joining node must be issued a membership certificate along with the secret share. By coupling BiAC with the threshold BLS signature scheme of Boldyreva [4] (which has a non-interactive signing procedure) to issue certificates to the joining nodes, we obtain a fully non-interactive admission protocol for long-lived MANETs. Evaluation of this extension is an item for future work.

8 Conclusion

In this paper, we considered two classes of MANETs: *short-lived* and *long-lived*, and showed that more efficient admission control protocols can be constructed for the former. We presented BiAC, an efficient and fully non-interactive admission control scheme based on bi-variate polynomial secret sharing. We demonstrated – via theoretical and experimental evaluation – that BiAC compares favorably to UniAC. (Recall that all previous admission protocols were originally designed for long-lived MANETs and are based on uni-variate polynomial secret sharing.)

[7] DL assumption: In a cyclic group generated by $g \in Z_p^*$ of order q, for $a \in \mathbb{Z}_q^*$, given (g, g^a), it is hard to compute a.

References

[1] K. Barr and K. Asanovic. Energy Aware Lossless Data Compression. In *ACM International Conference on Mobile Systems, Applications, and Services*, pages 231–244, May 2003.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *ACM Symposium on the Theory of Computing*, pages 1–10, May 1988.

[3] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. In *CRYPTO'92*, number 740 in LNCS, pages 471–48, Aug. 1999.

[4] A. Boldyreva. Efficient Threshold Signatures, Multisignatures and Blind Signatures based on the Gap-Diffie-Hellman-Group Signature Scheme. In *International Workshop on Practice and Theory in Public Key Cryptography*, number 2567 in LNCS, pages 31–46, 2003.

[5] G. Bracha. An Asynchronous $\lfloor (n-1)/3 \rfloor$-resilient Consensus Protocol. In *ACM Symposium on Priniciples of Distributed Computing*, pages 154–162, Aug. 1984.

[6] C. Castelluccia, N. Saxena, and J. H. Yi. Self-Configurable Key Predistribution in Mobile Ad Hoc Networks. In *IFIP Networking Conference*, number 3462 in LNCS, pages 1083–1095, May 2005.

[7] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. In *CRYPTO'89*, number 435 in LNCS, pages 307–315, Aug. 1990.

[8] P. Feldman. A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *IEEE Symposium on Foundations of Computer Science*, pages 427–437, Oct. 1987.

[9] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust Threshold DSS Signatures. In *CRYPTO'96*, number 1070 in LNCS, pages 354–371, May 1996.

[10] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. In *EUROCRYPT'99*, number 1592 in LNCS, pages 295–310, May 1999.

[11] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing, Or How To Cope With Perpetual Leakage. In *CRYPTO'95*, number 963 in LNCS, pages 339–352, Aug. 1995.

[12] Y.-C. Hu, D. B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 3–13, June 2002.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *ACM International Conference on Mobile Computing and Networking*, pages 12–23, Sept. 2002.

[14] S. Jarecki and N. Saxena. A Public Key Infrastructure based on Threshold Assumption. In *Submission*, July 2005.

[15] 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. In *ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 1–9, Oct. 2004.

[16] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *IEEE International Symposium on Network Computing and Applications*, pages 131–139, Apr. 2003.

[17] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu. Adaptive Security for Multi-level Ad-hoc Networks. In *Wiley Journal of Wireless Communications and Mobile Computing*, volume 2, pages 533–547, Aug. 2002.

[18] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for MANET. In *IEEE International Conference on Network Protocols*, pages 251–260, Nov. 2001.

[19] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *ACM Conference on Computers and Communication Security*, pages 52–61, Oct. 2003.

[20] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks. In *IEEE/ACM Transactions on Networking*, volume 12, pages 1049–1063, Dec. 2004.

[21] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing Ad Hoc Wireless Networks. In *IEEE Symposium on Computers and Communications*, pages 567–574, July 2002.

[22] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997. ISBN 0-8493-8523-7.

[23] G. Montenegro and C. Castelluccia. Crypto-based Identifiers (CBIDs): Concepts and Applications. In *ACM Transactions on Information and System Security*, volume 7, pages 97–127, Feb. 2004.

[24] M. Naor, B. Pinkas, and O. Reingold. Distibuted Pseudo-Random Functions and KDCs. In *EUROCRYPT'99*, number 1592 in LNCS, pages 327–346, May 1999.

[25] M. Narasimha, G. Tsudik, and J. H. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control. In *IEEE International Conference on Network Protocols*, pages 336–345, Nov. 2003.

[26] OpenSSL Project, http://www.openssl.org.

[27] R. Ostrovsky and M. Yung. How to Withstand Mobile Virus Attacks. In *ACM Symposium on Priniciples of Distributed Computing*, pages 51–61, Aug. 1991.

[28] Peer Group Admission Control Project. http://sconce.ics.uci.edu/gac.

[29] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1992. ISBN 0-521-43108-5.

[30] N. Saxena, G. Tsudik, and J. H. Yi. Admission Control in Peer-to-Peer: Design and Performance Evaluation. In *ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 104–114, October 2003.

[31] N. Saxena, G. Tsudik, and J. H. Yi. Identity-based Access Control for Ad-Hoc Groups. In *International Conference on Information Security and Cryptology*, number 3506 in LNCS, pages 362–379, Dec. 2004.

[32] A. Shamir. How to Share a Secret. In *Communications of the ACM*, volume 22, pages 612–613, Nov. 1979.

[33] M. Steiner, G. Tsudik, and M. Waidner. CLIQUES: A New Approach to Group Key Agreement. In *IEEE International Conference on Distributed Computing Systems*, pages 380–387, May 1998.

[34] M. Steiner, G. Tsudik, and M. Waidner. Key Agreement in Dynamic Peer Groups. In *IEEE Transactions on Parallel and Distributed Systems*, volume 11, pages 769–780, July 2000.

[35] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. In *IEEE Network Magazine*, volume 13, pages 24–30, Nov. 1999.

[36] L. Zhou, F. Schneider, and R. van Renesse. COCA: A Secure Distributed On-line Certification Authority. In *ACM Transactions on Computer Systems*, volume 20, pages 329–368, Nov. 2002.