Identity-based Access Control for Ad Hoc Groups*

Nitesh Saxena, Gene Tsudik, Jeong Hyun Yi**

School of Information and Computer Science
University of California at Irvine
Irvine, CA 92697, USA
{nitesh,gts,jhyi}@ics.uci.edu

Abstract. The proliferation of group-centric computing and communication motivates the need for mechanisms to provide *group access control*. Group access control includes mechanisms for admission as well as revocation/eviction of group members. Particularly in ad hoc groups, such as peer-to-peer (P2P) systems and mobile ad hoc networks (MANETs), secure group admission is needed to bootstrap other group security services. In addition, secure membership revocation is required to evict misbehaving or malicious members. Unlike centralized (e.g., multicast) groups, ad hoc groups operate in a decentralized manner and accommodate dynamic membership which make access control both interesting and challenging. Although some recent work made initial progress as far as the admission problem, the membership revocation problem has not been addressed.
In this paper, we develop an identity-based group admission control technique which avoids certain drawbacks of previous (certificate-based) approaches. We also propose a companion membership revocation mechanism. Our solutions are robust, fully distributed, scalable and, at the same time, reasonably efficient, as demonstrated by the experimental results.

Keywords: access control, ad-hoc group security, threshold signatures

1 Introduction

Ad hoc groups are becoming increasingly popular these days. A number of peer-to-peer (P2P) systems as well as mobile ad-hoc networks (MANETs) fall into the category of ad hoc groups. These groups are characterized by two important features, (1) lack of trusted authority and (2) dynamic membership, which often implies dynamic topology. These features prompt a number of challenges for routing as well as content placement and retrieval. They also make it difficult to develop effective and efficient security mechanisms. The need for security in

* This work was supported in part by an award from the Army Research Office (ARO) under contract W911NF0410280 and a grant from SUN Microsystems.

** corresponding author

MANETs and P2P has been widely recognized by the research community and the bulk of prior work has been in the context of traditional security services such as secure group communication (group key agreement and key management) and secure routing (in MANETs). Although these services are certainly important, another equally important issue – **group access control** – has not received due attention.

In a group setting, the traditional notion of access control is to prevent unauthorized entities from accessing group resources. However, if we consider group membership itself to be a resource, the problem of admission and revocation/eviction of group members can be viewed as a form of access control. A secure admission control is necessary to prevent unauthorized users from joining a group, i.e., accessing the "membership" resource. Without such a mechanism, group membership is open to malicious users and the group becomes vulnerable, e.g., to Sybil attacks [1]. Moreover, a group as a whole must be able to contend with the possibility of group members becoming selfish[1] or malicious/compromised. Once detected, such rogue members need to be removed from the group. This necessitates secure (and efficient) membership revocation mechanisms.

Without effective group access control mechanisms, other group security services: secure group communication [2, 3] and secure routing (in mobile ad hoc networks), such as Ariadne [4], SPINS [5], etc., are difficult to achieve.

One fairly obvious application for the type of secure distributed group access control that we envisage is in the domain of private P2P groups formed atop wide-open (essentially public) P2P systems, such as Kazaa, Morpheus or Gnutella. In fact, a recent article in the Time Magazine [6] examines the popular trend of creating so-called *"Darknets"* [7] – secure private groups within Kazaa and Morpheus – in order to escape the intensified crackdown on music and other content sharing. Another example of a somewhat futuristic, military oriented application of group access control in a mobile ad hoc network with unmanned aerial vehicles (UAVs) is described in [8].

2 Related Work and Motivation

Previous work on admission control in ad hoc groups ([9], [10], [11], and [12]) employed a menu of cryptographic techniques to perform secure group admission. The purpose is for a certain threshold of group members to make collaborative decisions regarding the admission of a prospective member and provide it with a signed group membership certificate. Among these signature schemes are: plain (RSA or DSA) signatures, accountable subgroup multi-signatures (ASM) [13], threshold signatures ([14], [9]). Unfortunately, these schemes have certain drawbacks that make them unfit for practical group admission control scenarios.

[1] A selfish member is interested in obtaining service but refuses to provide it. Service can range from sharing files in a content-sharing P2P to forwarding traffic in a MANET.

Lineage problem with plain signatures and multisignatures. In particular, admission control based on either plain signatures or accountable sub-group multisignatures has a *lineage* problem. This problem occurs when a membership certificate is issued to a new member: each member (sponsor) who takes part in the admission process needs to confirm (by signing) its agreement to admit this new member. Essentially, a membership certificate has to be signed by some number of membership sponsors.[2] However, each sponsor needs to attach its own certificate to its signature on a new member's certificate in order to make group certificates universally verifiable. However, a sponsor's own certificate also has to be counter-signed by its erstwhile sponsors, and so on, and so forth. This is clearly unworkable since a member's certificate would have to be accompanied by a number of certificate chains that affirm its lineage.

Inapplicability of known threshold RSA signatures. A $(t-1, n)$ threshold signature scheme [15] enables any subgroup of t members in a group to collaboratively sign a message on behalf of that group. This is achieved by secret-sharing the signature key among the group members, and allowing them to compute a signature on some message via a distributed protocol in which the members use the shares of the signature key instead of the key itself. Threshold signatures are naturally attractive for the purpose of admission control since they prevent the aforementioned lineage problem: a membership certificate can be signed by any set (of a certain size) of sponsors while the signature length is constant and identities of individual sponsors are not revealed.

Various flavors of threshold RSA signatures exist in literature that might be used to construct the admission control protocol. However, unfortunately, **none** of these schemes are directly applicable (refer to [16] for details).

In an effort to mitigate the above problem of the known threshold RSA signatures, Kong, et al. [9] proposed a new threshold RSA scheme, geared toward providing security services in mobile ad-hoc networks. Subsequently the scheme and its MANET applications were described in [9, 8], and most recently in a journal version [17]. Unfortunately, this scheme is neither robust (i.e. it can not tolerate malicious group members) nor secure. The robustness problem was first pointed out in [12]. For an explicit attack exploiting the insecurity of the scheme, the reader is referred to [18]

Limitation of threshold DSA signature. An alternative scheme (called *TS-DSA*) [12] is based on the threshold DSS signature scheme [19]. This scheme is robust and hence tolerates malicious insiders. However, the practicality of this scheme is questionable since, as illustrated by experiments in [11], it is very costly due to $O(t^2)$ communication among the t signers. Furthermore, *TS-DSA* remains secure as long as there are less than $\lfloor \frac{t+1}{2} \rfloor$ malicious members. In other words, for the scheme to be able to tolerate $t-1$ faults, $2t-1$ signers are required.

[2] This number is determined by the group admission policy; common examples are a certain fraction of current members or a fixed threshold. See [10] for a detailed discussion of admission policies.

Our Contributions. In addition to the above discussion, the previously proposed admission control mechanisms are certificate-based, making them quite impractical[3] for mobile ad hoc networks where the amount of communication is directly related to the battery power of the mobile devices (refer to [20]). In this paper, we develop a new group admission control technique, which we refer to as *ID-GAC* (**ID**entity-based **G**roup **A**dmission **C**ontrol). As the name suggests, *ID-GAC* is an identity-based approach (and therefore more communication efficient), in contrast to previous, certificate-based schemes. Furthermore, we present a membership revocation mechanism geared to work in conjunction with *ID-GAC*.

Organization. After specifying the notation in Table 1, we define the security model for a generic group access control system in Section 3. In Sections 4 and 5, we present the new ID-based admission control protocol and demonstrate its security. Section 6 presents the membership revocation protocols. Experimental results are illustrated in Section 7 and finally, some outstanding issues are discussed in Section 8 followed by the conclusion in Section 9.

Table 1. Notation

M_i	group member i	id_i	ID for M_i
t	admission threshold	t_r	revocation threshold
n	total number of M_i-s	$\mathbb{G}_1, \mathbb{G}_2$	cyclic GDH groups of order q
A	generator of group $\mathbb{G}_1$	B	group public key
$\hat{e}$	map s.t. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$	T_i	membership token for M_i
SK_i	secret key of M_i	PK_i	public key of M_i
$S_i(m)$	signature on message m	MRL	membership revocation list
SL_i	list of signers for M_i	H	hash func. such as SHA-1 or MD5
ss_i	secret share of M_i	H_1	hash func. s.t. $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$
$pss_j(i)$	partial share for M_i by M_j	H_2	hash func. s.t. $H_2 : \{0,1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$

3 Security Model

In this section, we define a generic security model for peer-based membership management. In other words, we describe what we mean by secure admission control and secure membership revocation, respectively. Since this work is applied in nature, the model is specified informally.

3.1 Admission Control

A *secure admission* mechanism is a secure interactive protocol between a prospective member M_{new} and a set of current group members $\{M_i \mid 1 \leq i \leq s\}$ where

[3] The typical size of a group membership certificate is quite large, e.g., 5KB for 1024-bit DSA parameters.

$1 \leq t \leq s \leq n$. (In other words, the number of current members taking part in the admission is at least the number necessary for the admission threshold which is, in turn, no greater than the number of current members.) This protocol must satisfy the following properties:

1. Completeness. When the protocol completes (in polynomial time), M_{new} has a group membership token, if at least t out of n group members vote in favor of admission. In addition, M_{new} also acquires the membership revocation information if any, which allows it to keep track of revoked group members. *Optionally*[4], M_{new} is also in possession of a means to take part in future admission decisions.
2. Traceability. If one or more malicious sponsors do not provide correct information in the course of the admission process, M_{new} can detect, trace and publicly identify such members.[5]
3. Impersonation Resistance. It is computationally infeasible for anyone, who has not been successfully admitted via the admission protocol, to impersonate a genuine group member (whether to members or non-members).

3.2 Membership Revocation

A secure *membership revocation* mechanism is a protocol among the members of the group wherein any set of t_r ($\geq t$) members attempt to revoke a current member M_r who possesses a membership token T_r and/or a secret share ss_r. This mechanism needs to satisfy the following properties:

1. Completeness. At the end of this protocol, i.e. when t_r revocation requests are lodged against M_r, the latter is unable to: (1) prove group membership and (2) participate in future admission decisions (in case M_r previously had voting/admission rights).
2. Impersonation Resistance. Only a genuine group member (possessing a valid membership token) can take part in the revocation process. In other words, non-group members and previously revoked group members can not lodge valid revocation requests.
3. Collusion Resistance. At the end of the protocol, M_r (if she possessed a secret share) is unable to collaborate with any set of previously revoked members and recover the group secret x. In other words, any set of revoked share-holding members can not collude to interpolate the group secret.

Remark: As will be discussed in Section 6, every group member maintains an updated list of the revoked members which we refer to as *membership revocation list* (MRL).

[4] Whether a new member gets voting rights that allow it to take part in future admission decisions depends upon group admission policy.

[5] The group members misbehaving in this manner may then be revoked from the group by triggering the revocation mechanism.

4 ID-based Group Admission Control (ID-GAC)

In this section, we present our new admission control mechanism *ID-GAC*. The mechanism is based on the threshold version [21] of BLS signature scheme [22]. The description includes: set-up, admission process and security arguments following the model in Section 3. *ID-GAC* is an identity-based mechanism since the membership token used to prove membership is derived from the group member's identity.

4.1 Setup

ID-GAC can be initialized by either: (1) a trusted dealer or (2) a group of $2t-1$ or more founding members. In either case, the dealer first initializes and generates the appropriate elliptic curve domain parameters $(p, \mathbb{F}_p, a, b, A, q)$. The elliptic curve is represented by the equation: $y^2 = x^3 + ax + b$. $\mathbb{G}_1$ is set to be a group of order q generated by A, $\mathbb{G}_2$ is a subgroup of $\mathbb{F}^*_{p^2}$ of order q, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined to be a public bilinear mapping. Also, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ is the hash function that maps binary strings to non–zero points in $\mathbb{G}_1$. All of this information is published and all group members (as well as prospective members) are assumed to have access to it.

INITIALIZATION BY DEALER. TD selects a random polynomial $f(z) = f_0 + f_1 z + \cdots + f_{t-1} z^{t-1}$ over $\mathbb{Z}_q$ of degree $t-1$, such that the group secret is $f(0) = f_0 = x$. In order to enable verifiable secret sharing (VSS) [23], TD computes and publishes the witnesses $W_i = f_i A$ for $(i = 0, \cdots, t-1)$. The witness value $W_0 = xA$, also denoted by B, is actually the group public key. Next, for each M_i, TD computes the secret share ss_i and the identity-based membership token T_i (valid until the time exp[6]) such that: $ss_i = f(id_i) \pmod q$ and $T_i = xH_1(id_i||exp)$. Note that TD is not required hereafter.

SELF-INITIALIZATION BY FOUNDING MEMBERS. t or more founding members M_i-s select individual polynomials $f_i(z)$ over $\mathbb{Z}_q$ of degree $t-1$, such that $f_{i0} = x_i$. Then, using the DKG protocol [24], each M_i computes its own secret share ss_i, such that $ss_i = \sum_{j=1}^{l} f_j(id_i) \pmod q$ $(l \geq 2t-1)$. Once M_i gets its share, it is rather easy to recover the secret using Lagrange interpolation. Also, the dealing process supports VSS. Now, in order to provide each member with a membership token, any set of t founding members must collaborate. For example, group

[6] Membership tokens are valid for a certain period of time. The duration of the validity period are be defined by the group policy (which is out of scope of this paper). For simplicity, we assume that all membership certificates reflect the same expiration period exp. In order to enable implicit revocation, each member M_i needs to be provided with $T_i = xH_1(id_i||exp)$. This structure implies that T_i is not valid after the time exp. Once expired, the token needs to be renewed via the admission process. The group founding members might be provided with long(er)-term membership tokens. We assume that all nodes have reasonably synchronized clocks, within a certain skew.

members $M_2, M_3, \cdots, M_{t+1}$ may collaborate and provide M_1 with a membership token as $T_1 = \sum_{j=2}^{t+1}(ss_j l_j(0))H_1(id_1||exp)$ $[= xH_1(id_1||exp)]$.

4.2 Admission Process

Let n $(\geq t)$ be the number of current group members. In order to be admitted to the group, a prospective member M_{new} must collect at least t votes from current group members. Figure 1 shows protocol message flows for the admission process. The goal is for M_{new} to obtain a membership token T_{new} which can then be used to prove membership. (In practice, M_{new} also needs to obtain the current membership revocation list – MRL – to keep track of revoked group members.)

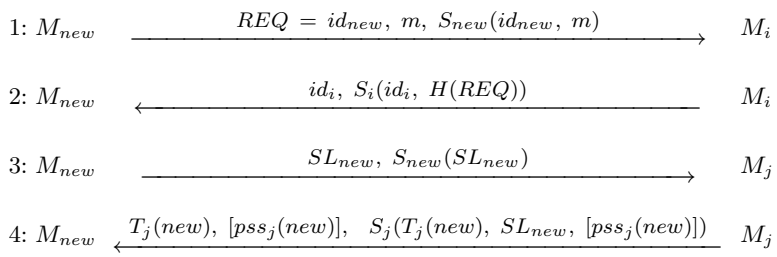


Fig. 1. *ID-GAC* Protocol

1. M_{new} sends a signed join request message m as well as its identity id_{new} to at least t current group members[7].
2. Group members who wish to participate in the admission reply with their respective id-s to M_{new} along with the signature on the previous message sent by M_{new}.
3. M_{new} picks (perhaps, at random) t sponsors M_j-s, forms a signer list SL_{new} which contains the id-s of t responders, signs it, and sends it to each M_j.
4. Each signing member M_j sends back to M_{new} the partial membership token $T_j(new)$ and (optionally) the partial share of the secret $pss_j(new)$ such that

$$T_j(new) = (ss_j \cdot l_j(0))H_1(id_{new}||exp) \quad \text{and}$$
$$pss_j(new) = ss_j \cdot l_j(id_{new}) + r_j \pmod q$$
$$\text{where, } l_j(x) = \prod_{i=1}^{t} \frac{x - id_i}{id_j - id_i}.$$

   M_{new} is also provided with a signed copy of the current MRL. Note that the Lagrange coefficients $l_j(id_{new})$-s are publicly known, and therefore, M_{new}

[7] In order to secure the protocol against common *replay* attacks [25], we note that it is necessary to include timestamps, nonces and protocol message identifiers. However, in order to keep our description simple, we omit these values.

can derive ss_j from $pss_j(new)$. This can be prevented using the *shuffling* technique proposed in [9] by adding extra random value r_j-s to each share. These r_j-s are secret random values and must sum up to zero by construction. They must be securely shared among the t sponsoring M_j-s.

5. Finally, M_{new} calculates its secret share ss_{new} (if provided) and the membership token T_{new} by adding up the values obtained in the last step. The share acquisition and membership token acquisition procedures are discussed next.

4.3 Membership Token Acquisition

The membership token acquisition procedure is also performed as part of the 4-th message of the above protocol. Each sponsor M_j computes a partial membership token $T_j(new)$ for M_{new} and then sends it to M_{new}. (It is important to note that the partial membership token $T_j(new)$ is actually a BLS [22] signature on id_{new} using $(ss_j \cdot l_j(0))$ as the secret key for M_j.) Then, M_{new} computes its membership token T_{new} by summing up $T_j(new)$ $(j = 1, \cdots, t)$ and verifies the correctness by checking $\hat{e}(B, H_1(id_{new}||exp)) = \hat{e}(A, T_{new})$. This equation can be easily shown to be correct using the properties of the bilinear map $\hat{e}$.

4.4 Share Acquisition

The share acquisition procedure – whereby M_{new} obtains its share ss_{new} from sponsors – is performed as part of the 4-th message in the *ID-GAC* protocol (as shown in Figure 1).

The t sponsoring members (M_j-s) compute the shuffled partial share $pss_j(new)$ for M_{new} as $pss_j(new) = ss_j \cdot l_j(id_{new}) + r_j \pmod q$ using the shuffling technique [9]. Each M_j sends $pss_j(new)$ to M_{new}. Then, M_{new} computes its share ss_{new} by summing up $pss_j(new)$ $(j = 1, \cdots, t)$ and verifying the correctness using VSS [23].

By verifying the secret share and the membership token as described above, M_{new} is assured of possessing correct credentials. Armed with the membership token T_{new}, M_{new} can prove membership. Also, using a secret share ss_{new}, M_{new} can use the group secret x in collaboration with other $(t-1)$ group members and take part in future admission protocols. In Section 5, we discuss two schemes that can be used to prove membership.

Next, we discuss the security of the proposed *ID-GAC* scheme.

4.5 Security Considerations

In this section, we argue the security of the proposed scheme, based on the security model of Section 3.

1. Completeness. This property follows by inspection. At the end of the protocol, M_{new} receives the membership token T_{new} which is verified as:

$\hat{e}(B, H_1(id_{new}||exp)) = \hat{e}(A, T_{new})$. Using T_{new}, M_{new} can prove membership. M_{new} also receives a secret share ss_{new} which is verified using VSS as: $ss_{new}A = \sum_{i=0}^{t-1} {id_{new}}^i W_i$. Using ss_{new}, M_{new} can take part in future admission decisions and can also recover the group secret x in collaboration with any other $t-1$ members. Of course, M_{new} can obtain these credentials in polynomial time.

2. TRACEABILITY. In case the verification of T_{new} and/or ss_{new} fails, M_{new} must identify (trace) sponsors that sent invalid partial token(s) and/or partial secret share(s). To verify each partial secret share, M_{new} can perform the VSS procedure. Correctness of each membership token share $T_j(new)$ can be verified as follows:

$$\hat{e}(T_j(new), A) = \hat{e}(H_1(id_{new}||exp), l_j(0) \sum_{i=0}^{t-1} {id_j}^i W_i).$$

   If the above verification fails, M_{new} concludes that M_j is cheating.
3. IMPERSONATION RESISTANCE. In order to make sure that M_{new} is communicating with only genuine group members, it can verify the partial credentials as in TRACEABILITY above and in the process can trace any impersonating non-member.

5 Proving Membership

We employ two schemes that can be used by a group member to prove membership. We consider proving membership to internal parties (members) and external parties (non-members). The internal membership proof (IMP) is a pairing-based secret handshake scheme proposed in [26]. The external membership proof (EMP) is the identity-based signature scheme in GDH groups as proposed in [27].

6 Membership Revocation

In addition to implicitly revoking membership tokens via expiration (see footnote 6), rogue members need to be explicitly revoked, e.g., for reasons of selfishness, maliciousness or compromise.

A secure membership revocation mechanism should satisfy all properties outlined in Section 3.2. One trivial solution is to have a set of t share holding group members (revokers) collaborate and renew membership tokens for all members, except the one being revoked. The revokers also need to update the secret shares using proactivity in case the revoked member possessed an old share. This approach is clearly very inefficient.

In this section we present a practical revocation mechanism based on Membership Revocation Lists (MRLs). This is analogous to the simple and widespread certificate revocation technique (CRLs [28]) used in traditional PKIs. However, unlike CRLs, our solution is fully distributed.

6.1 MRL Update

Upon every revoke operation, each group member needs to update its copy of the MRL. Also, an entry needs to be removed from the MRL once the corresponding member's membership token expires.

6.2 Membership Validation

If and when a user V receives a signed message from another user M_u claiming to be the group member (of a group with public key B), V needs to first verify the validity of M_u's membership and then the signature (using EMP signature verfication). The validation procedures will be different (internal or external) based on whether V is a group member or a non-member.

Internal Membership Validation. If V is a group member, validation involves only a lookup of its MRL and checking the status of M_u. If MRL contains no entry corresponding to id_u, V concludes that M_u is a member in good standing.

External Membership Validation. An non-member V needs to perform the following protocol to validate M_u's status.

1. V sends to the group a membership validation request for M_u's membership.
2. Share holding group members interested in answering this request, reply with their IDs to V.
3. V waits for at least t such responses, creates a signers' list SL_v containing the IDs of the interested members and sends it to them.
4. These members then look up the status of M_u in their respective MRLs and provide V with a signed response.
5. V verifies the signature on the status of M_u.

The above validation procedure is quite similar to the admission process described in Section 4.2. Moreover, it could also be viewed as a distributed version of the Online Certificate Status Protocol ($OCSP$) [29] in the context of certificate revocation.

6.3 Revocation Process

To revoke an allegedly malicious or misbehaving member M_r, any current member M_a can bootstrap the revocation process. The following steps need to be performed.

1. M_a broadcasts a revocation request message referencing M_r, using EMP signature generation.

2. All other group members M_j-s ($j \neq r$) perform the internal membership validation for M_a as in Section 6.2. [8] They then perform the MRL update (as discussed in Section 6.1) to add M_a to the revoker list for M_r. The status of M_r is then set to "under-review". Once the number of revokers in this list reach t_r, the status of M_r is updated to "revoked".
3. If M_r is the $\hat{t}^{(th)}$ ($\hat{t} \leq t-1$) member to be revoked[9], the t_r revokers collaborate and update the shares using the proactive method. If less than t out of t_r revokers possessed the secret shares, the group founding members need to perform the share update.
4. All available group members M_i-s ($i \neq r$) possessing voting rights, then contact the revokers (or founding members) and renew their shares.

6.4 Security Considerations

In this section, we argue that the above revocation mechanism is secure based on the security model sketched in Section 3.2.

1. Completeness. By inspection: As soon as t_r valid revocation requests are lodged against M_r, each group member records them in its local copy of the MRL and sets the status corresponding to id_r as "revoked". Now, M_r can not prove membership via either IMP or EMP protocol and/or take part in the admission process since its membership validation (as described in Section 6.2) would fail.
2. Impersonation Resistance. As part of Step 1 of the revocation process above, a revocation request submitted by M_a against M_r is signed using EMP signing. Upon receiving such a request, each group member (except M_a) first validates M_a's status by doing an MRL lookup and then validates M_a's membership by verifying the signature on the request message (via EMP verification). The former guarantees that M_a is itself not revoked and the latter ensures that M_a is indeed a group member. Therefore, it is impossible for a revoked member and computationally infeasible for a non-member to lodge valid revocation requests. Thus, the proposed revocation mechanism is impersonation resistant.
3. Collusion Resistance. The revocation procedure involves the secret share updates atleast after every $t-1$ members are revoked. Therefore, the share of the t^{th} revoked member will not correspond to the shares of $t-1$ previously revoked members in yielding the group secret using the polynomial interpolation. This implies that no set of revoked members can collude.

6.5 Discussion

The MRL-based solution requires group members to synchronize in order to maintain up-to-date MRLs. However, it is certainly unrealistic to expect all

[8] Although this is concerned with the group revocation policy, here we assume that a member which is "under-review" can also lodge valid revocation requests.

[9] How many revocation operations trigger a share update is determined by the group revocation policy.

members to be on line all of the time. Any member can establish the freshness of its MRL by performing a procedure similar to membership validation (in Section 6.2). A set of t interested members (possessing the secret shares) respond with the complete and signed MRL, as opposed to the membership validation procedure, where they respond with the status of a particular member. This procedure can also be performed periodically.

Discrepancies in MRLs might arise for a number of reasons. A group member might go off-line or become unreachable temporarily. Such events are common in asynchronous groups, such as most P2P systems and MANETs. More generally, a group can become partitioned due to some network event, e.g., a router failure. Suppose a partition occurs and a group is split into two subgroups G_A and G_B. The two subgroups operate independently and their MRLs evolve separately. If at a later time, G_A and G_B merge back into a single group, the two MRLs: MRL_A and MRL_B, need to be appropriately merged. This particular scenario presents a major challenge since the techniques described above will not work. Consider what happens if, while the group is partitioned, members of G_A decide to revoke their counterparts in G_B, and vice versa. (Not surprisingly, this remains a major item for future work.)

7 Performance Analysis

We now present and discuss the performance measurement results for the proposed *ID-GAC* admission and eviction techniques. In particular, we describe our experience with the implementation of these schemes and experiments in P2P and MANET settings, focusing on the respective costs of admission, traceability, membership proofs and revocation. We also compare our results with the previously proposed DSA-based admission mechanism [12], [11], wherever applicable.

7.1 Implementation

The *ID-GAC* library is built using OpenSSL [30] and MIRACL [31] (optimized using Comba method) libraries. The latter was needed to implement various identity-based functions. Currently, *ID-GAC* consists of approximately $10,000$ lines of C/C++ source code and supports Linux 2.4.

We used the elliptic curve E defined by the equation: $y^2 = x^3 + 1$ over $\mathbb{F}_p$ with $p > 3$ a prime satisfying $p = 2 \ (mod)\ 3$ and q being a prime factor[10] of $p + 1$. The size of q is set to be 160 bits and p is a 512-bit prime. The group $\mathbb{G}_1$ is a subgroup of points generated by A such that $A \in E(\mathbb{F}_p)$. The group $\mathbb{G}_2$ is a subgroup of $\mathbb{F}^*_{p^2}$ of order q. The bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is the well-known Tate pairing. Note that the pairing value belongs to finite field of 1024 bits.

[10] By Euler's theorem, q must divide $\#E(\mathbb{F}_p)$. For the curve $y^2 = x^3 + 1$, $\#E(\mathbb{F}_p) = p + 1$.

7.2 Basic Operations

To estimate the performance of *ID-GAC*, we first present the costs of the primitive operations in Table 2. For measuring the costs of basic operations in *ID-GAC*, we used a machine with an Intel P3 800MHz processor and 384MB memory. All experiments were repeated $1,000$ times for each measurement in order to get fairly accurate average results.

Table 2. Costs of Primitive Operations (P3-800MHz)

Function	modulus (bits)	exponent (bits)	average time (msec)
DSA sign	1024	160	4.78
DSA verify	1024	160	5.74
Map-to-point ($H_1(\cdot)$)	512	160	3.13
BLS sign	512	160	8.91
Pairing	512	160	37.24

7.3 Experimental Setup

We now describe the experimental setup used for the experiments in both P2P (Gnutella [32]) and MANET settings. We used two laptops and two PDAs, each configured with 802.11*b* in ad-hoc mode. For routing purposes, we used the Optimized Link State Routing Protocol (OLSR) [33]. The two laptops (running Linux 2.4) are equipped with 800/900-MHz P-III processors and 384/256MB RAM, respectively. The two PDAs (running Linux *Familiar*) each have a 400MHz XSCALE processor and 64MB RAM. In all experiments, we ran equal number of processes (current group members) on both the laptops. The PDAs were used for routing purposes only.

Note that both P2P and MANET experiments were done on the equipment with same computing power. The main purpose of our experiments is to measure the computation and communication costs in wireless as well as wired networks and to demonstrate the practicality of ID-based admission and revocation mechanisms. For *ID-GAC* experiments, the modulus size ($|p|$) was set to 512-bits and 1024-bit modulus was used for *TS-DSA* experiments[11].

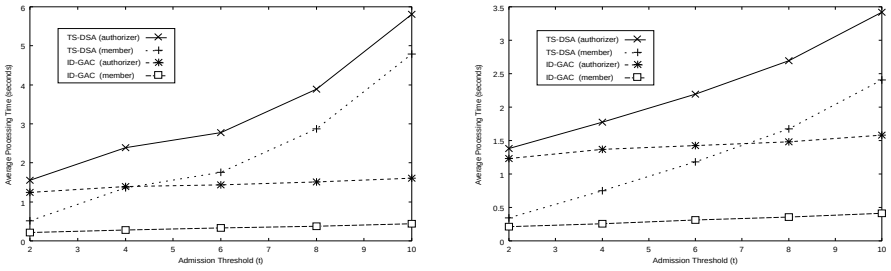
Remark: In all experiments below, we used a member/authorizer paradigm. A *member*, in this context, is a group member who has no voting/admission rights (i.e., no secret share), whereas an *authorizer* has them. Since their respective costs sometimes vary substantially, they are graphed separately.

[11] Computing discrete log in $\mathbb{F}_{p^2}$ is sufficient for computing discrete log in $\mathbb{G}_1$. Therefore, for proper security of discrete log in $\mathbb{F}_{p^2}$ the prime p should be at least 512-bits long (so that the group size is at least 1024-bits long). This will ensure that the GDH problem remains sufficiently hard.

7.4 Node Admission

Figure 2 shows the admission cost with varying threshold (for both member and authorizer) for *ID-GAC* and *TS-DSA* schemes in (a) MANET and (b) Gnutella experiments. The admission costs also include the verification of the membership token and/or the secret share.

As shown in Figure 2, *ID-GAC* exhibits appreciably better performance that *TS-DSA* in MANET, as well as in P2P setting. The results imply that the amount of communication in *TS-DSA* contributes significantly to the overall cost of admission, although computation-wise it is still quite efficient (see Table 2). The communication overhead for *TS-DSA* is even higher in MANET, than in Gnutella experiments. This is clearly due to the error-prone, low-bandwidth wireless channel.



(a) MANET (b) Gnutella

Fig. 2. Node Admission Cost

7.5 Bandwidth Consumption

Table 3 compares the respective bandwidth costs. (Refer to Table 1 for notation.) While *TS-DSA* uses certificates, *ID-GAC* is identity-based which obviates any need for explicit membership certificates. Certificate size is relatively large, e.g., 5KB with 1024-bit DSA parameters. For example, if a prospective member wants to join the group as a member, $(2t-1)*5K*8$ bits must be transfered, whereas, only $t*512$ bits are needed in *ID-GAC*.

Table 3. Bandwidth Comparison

	TS-DSA	ID-GAC
Admission (member)	$(2t-1)*\|GMC_j\|$	$t*\|T_j\|$
Admission (authorizer)	$(2t-1)*\|GMC_j\|+t*\|pss_j\|$	$t*(\|T_j\|+\|pss_j\|)$

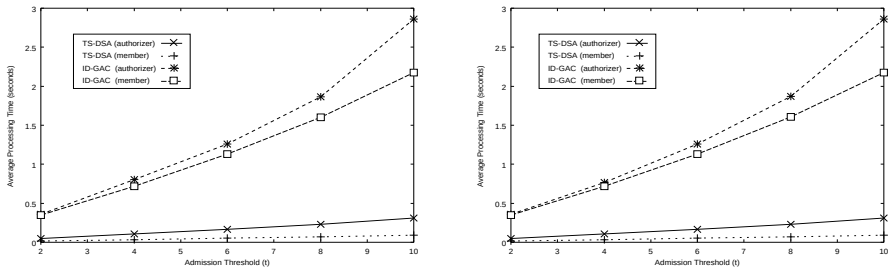
Also, it is well known that, in many small devices (such as low-end MANET nodes or sensors) sending a single bit is roughly equivalent to adding $1,000$ 32-

bit numbers, in terms of battery power consumption [20]. For example, in case of $t = 3$, the bandwidth cost with *TS-DSA* is about 133 times higher than *ID-GAC*. In other words, *TS-DSA* consumes 133 times more energy than *ID-GAC* for the communication. Hence, we expect *ID-GAC* to be more suitable for MANET scenarios which often involves power-constrained devices.

7.6 Traceability

Traceability costs are presented in Figure 3. Due to the costly computation of Tate pairings, *ID-GAC* performs poorly, as compared to *TS-DSA*.

However, since the misbehavior in the admission protocol leads ultimately to the eviction of the corresponding group member, we argue that traceability is a *rare exceptional* measure; thus we consider its costs to be relatively unimportant.



(a) MANET (b) Gnutella

Fig. 3. Traceability Cost

7.7 Membership Proof

We now discuss measurement results for the membership proofs in *ID-GAC*, as outlined in Section 5. The respective costs of IMP and EMP computations (including both signing and verifying) are shown in Figure 4, with varying key sizes. Recall that IMP is needed if two members want to authenticate each other secretly and establish a shared secret key; in contrast, EMP can (also) be used to prove membership to outsiders.

7.8 Revocation

The graph in Figure 5 represents the average cost needed to revoke a particular group member for the varying threshold using the mechanism described in Section 6. In this context, the threshold (t_r) is the revocation threshold. Costs include: generation of signed revocation requests, broadcast to the group and validation of these requests. Delay between two consecutive revocation requests is not taken into account.

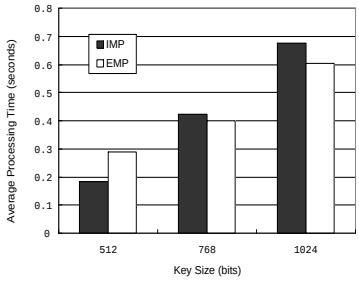


Fig. 4. Membership Proof Cost

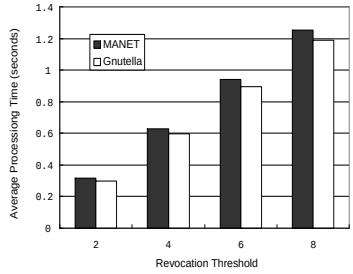


Fig. 5. Membership Revocation Cost

8 Discussion and Further Improvement

In this section, we discuss some of the outstanding issues concerning *ID-GAC*, focusing primarily on the performance.

The bilinear mapping operation is an elegant procedure that forms the basis for the verification in the admission process and the membership proof. However, it is an expensive computation which dominates the overall running time of the protocol. Moreover, it is only because of the Tate pairing operation in our protocol that the key size, i.e. the size of the prime p needs to be at least 512-bits; although it is well-known that a 160-bits ECC system is as secure as a 1024-bits RSA based system. Barreto, et al. [34] suggest some modifications and optimizations for the Tate pairing operation, most of which are implemented in the MIRACL library [31] that we used. In order to further improve performance, we need to parallelize these operations and pre-compute as much as possible. As discussed in Section 4.5, the traceability of malicious sponsors is an optional procedure that involves verification of individual sponsor's signatures, which costs two pairings per verification. This cost can be significantly lowered by using pre-computations as described in [31].

The choice of the elliptic curve being used can certainly influence the overall cost of the scheme. BLS short signature scheme [22] uses a supersingular curve defined by the equation $y^2 = x^3 + 2x \pm 1$ over $\mathbb{F}_{3^l}$ where l is a positive exponent. Barreto, et al. [34] specify the cost of generating a BLS signature on such a curve (defined in $\mathbb{F}_{3^{97}}$) to be just 3.57 ms on a P3 1GHz machine, after some optimizations and preprocessing. This seems like a significant improvement over the costs incurred in our measurements which were based on a different curve defined in a prime field. Therefore, porting our protocol for these curves appears to be an attractive way to reduce costs. Since the signature generation here is cheaper though the verification is still costly, this in fact could be an ideal candidate for an admission control mechanism, where one would prefer the computation load to be high on the prospective member, much lesser so on the current group members. In addition, these signatures are very short, only 154-bits in length,

which will give rise to short membership tokens in *ID-GAC*; another seemingly attractive prospect.

9 Conclusion

In this paper, we proposed *ID-GAC*, an identity-based scheme for secure admission control in dynamic ad hoc groups along with a distributed membership revocation mechanism based on the membership revocation lists. *ID-GAC* borrows ideas from threshold secret sharing and ID-based cryptography. As demonstrated by extensive experimentation in both P2P and MANET settings, the proposed scheme is far more efficient than the previously proposed solution, *TS-DSA*[11], [12]. The measurement results and performance analysis further indicate that *ID-GAC* is even more applicable in MANET devices where bandwidth and battery power are of prime concern.

References

1. Douceur, J.R.: The Sybil Attack. In: International Workshop on Peer-to-Peer Systems (IPTPS'02). (2002)
2. Steiner, M., Tsudik, G., Waidner, M.: Key Agreement in Dynamic Peer Groups. In: IEEE Transactions on Parallel and Distributed Systems. (2000)
3. Steiner, M., Tsudik, G., Waidner, M.: Cliques: A new approach to group key agreement. IEEE Transactions on Parallel and Distributed Systems (2000)
4. Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. In: Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002). (2002)
5. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: Spins: Security protocols for sensor networks. In: Mobile Computing and Networking. (2001)
6. Hamilton, A.: Playing in the dark: The heat is on, and music swappers are taking their business underground. TIME Magazine (2003)
7. Biddle, P., England, P., Peinado, M., Willman, B.: The darknet and the future of content distribution. In: ACM Workshop on Digital Rights Management. (2002)
8. Kong, J., Luo, H., Xu, K., Gu, D.L., Gerla, M., Lu, S.: Adaptive Security for Multi-level Ad-hoc Networks. In: Journal of Wireless Communications and Mobile Computing (WCMC). Volume 2. (2002) 533–547
9. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing Robust and Ubiquitous Security Support for MANET. In: IEEE 9th International Conference on Network Protocols (ICNP). (2001)
10. Kim, Y., Mazzocchi, D., Tsudik, G.: Admission Control in Peer Groups. In: IEEE International Symposium on Network Computing and Applications (NCA). (2003)
11. Saxena, N., Tsudik, G., Yi, J.H.: Admission Control in Peer-to-Peer: Design and Performance Evaluation. In: ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN). (2003) 104–114
12. Narasimha, M., Tsudik, G., Yi, J.H.: On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control. In: IEEE 11th International Conference on Network Protocol (ICNP). (2003) 336–345
13. Ohta, K., Micali, S., Reyzin, L.: Accountable Subgroup Multisignatures. In: ACM Conference on Computer and Communications Security. (2001) 245–254

14. Gennaro, R., S.Jarecki, H.Krawczyk, T.Rabin: Robust Threshold DSS Signatures. In Maurer, U., ed.: EUROCRYPT '96. Number 1070 in LNCS, IACR (1996) 354–371
15. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In Brassard, G., ed.: CRYPTO '89. Number 435 in LNCS, IACR (1990) 307–315
16. Saxena, N., Tsudik, G., Yi, J.H.: Access Control in Ad Hoc Groups. In: International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P). (2004)
17. Luo, H., Kong, J., Zerfos, P., Lu, S., Zhang, L.: URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, available on-line at `http://www.cs.ucla.edu/wing/publication/publication.html`. In: IEEE/ACM Transactions on Networking (ToN), to appear. (2004)
18. Jarecki, S., Saxena, N., Yi, J.H.: 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 (SASN). (2004)
19. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust and Efficient Sharing of RSA Functions. In Koblitz, N., ed.: CRYPTO '96. Number 1109 in LNCS, IACR (1996) 157–172
20. Barr, K., Asanovic, K.: Energy Aware Lossless Data Compression. In: International Conference on Mobile Systems, Applications, and Services (MobiSys). (2003)
21. Boldyreva, A.: Efficient threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Proceedings of International Workshop on Practice and Theory in Public Key Cryptography. Volume 2567 of LNCS. (2003) 31–46
22. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In Boyd, C., ed.: ASIACRYPT'01. Number 2248 in LNCS, IACR (2001) 514–532
23. P.Feldman: A Practical Scheme for Non-interactive Verifiable Secret Sharing. In: 28th Symposium on Foundations of Computer Science (FOCS). (1987) 427–437
24. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure Distributed Key Generation for Discrete-Log based Cryptosystems. In: Eurocrypt 99. (1999)
25. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press series on discrete mathematics and its applications. (1997) ISBN 0-8493-8523-7.
26. Balfanz, D., Durfee, G., Shankar, N., Smetters, D., Staddon, J., Wong, H.C.: Secret Handshakes from Pairing-Based Key Agreements. In: IEEE Symposium on Security and Privacy. (2003) 180–196
27. Cha, J., Cheon, J.: An ID-based signature from Gap-Diffie-Hellman Groups. In: Proceedings of International Workshop on Practice and Theory in Public Key Cryptography. Volume 2567 of LNCS. (2003) 18–30
28. Kocher, P.C.: On Certificate Revocation and Validation. In: Proceedings of International Conference on Financial Cryptography. Volume 1465 of LNCS. (1998)
29. Myersand, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: Online Certificate Status Protocol - OCSP . RFC 2560, IETF (1999)
30. OpenSSL Project: (`http://www.openssl.org/`)
31. MIRACL Library: (`http://indigo.ie/~mscott/`)
32. The Gnutella Protocol Specification v0.4: (`http://www.clip2.com/GnutellaProtocol04.pdf`)
33. OLSR: (`http://hipercom.inria.fr/olsr/`)
34. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient Algorithms for Pairing-based Cryptosystems. In Yung, M., ed.: CRYPTO '02. Number 2442 in LNCS, IACR (2002) 354–369