Admission Control in Collaborative Groups

Yongdae Kim[1], Daniele Mazzocchi[2], and Gene Tsudik[3]

[1] Department of Computer Science,
University of Minnesota, Twin Cities,
Minneapolis, MN
kyd@cs.umn.edu
[2] Department of Computer Science,
Politecnico Di Torino, Italy
dmazzocc@athena.polito.it
[3] Department of Information and Computer Science,
University of California at Irvine, CA, USA
gts@ics.uci.edu

Abstract. Security in collaborative peer groups is an active research topic. Most previous work focused on key management without addressing an important pre-requisite: admission control, i.e., how to securely admit a new member. This paper represents an initial attempt to sketch out an admission control framework suitable for different flavors of peer groups and match them with appropriate cryptographic techniques and protocols. Open problems and directions for future work are identified and discussed.

1 Introduction

The current proliferation of group-oriented applications, protocols and services triggers the need for specialized group security services and mechanisms. Examples of popular group-oriented settings include: IP telephony, video/audio conferencing, file sharing, collaborative workspaces, and multi-user games. Group settings are very diverse. Some, such as conferencing, require synchronous operation while others, such as Gnutella-style [20] file sharing, operate in disconnected, asynchronous manner. Communication models vary as well: from the one-to-many or few-to-many (e.g., GPS [19]) to any-to-any peer groups (e.g., Gnutella).

The need for, and the importance of, group security mechanisms has been recognized by the security research community. In the recent years, this topic has become quite popular. However, the bulk of prior work has been done in the context of large multicast-style groups with one or few senders and a multitude of receivers. In this setting, it is natural to assume or impose a centralized authority (be it the sender or an on-line trusted third party) that can perform many security chores, e.g., key management, admission/access control and member authentication. Such an authority may be group-specific or group-independent and its existence makes it relatively easy to implement security policies and mechanisms [24, 23, 27]. However, due to their peer nature, some group settings exhibit unique properties and requirements.

In this paper we focus on admission control mechanisms for peer groups. As the name suggests, a *peer* group is characterized by a flat structure meaning that there is no hierarchy among members and all members have identical rights and duties. In other words, there is no underlying assumption of a centralized authority who can provide security services such as access control or key management. Also, as mentioned above, a peer group usually involves any-to-any communication meaning that any member can send data to any other member or members. (In some cases, any member can communicate to the entire group.) As a result of these features, security in peer groups presents a formidable challenge. Lack of centralized authority typically entails the involvement of **all** group members in tasks such as key management. As evidenced by prior work in peer group key management, it is very hard to design multi-party, multi-round protocols that are – at the same time – secure, efficient and robust [5, 6, 4].

Although an important issue, peer group admission control has been somehow overlooked in the past. With the exception of Antigone project [27], most prior work in peer group security has focused on key management and authentication, whereas, without admission control, key management alone is all but useless. Consequently, our goal is to start by developing a framework for peer group admission control and, in doing so, analyze and propose a set of cryptographic mechanisms suitable for different peer group flavors.

Scope and Limitations: The work described in this paper is concerned only with peer group admission control and does not address the more general (and important) topic of group security policy. In the following, we assume the existence of group policy and do not deal with specification and negotiation of group policy.

Furthermore, in an effort to keep our contribution general, we avoid, wherever possible, mentioning the particulars of the peer group environment. This includes, but it not limited to features such as: network protocols, operating system details, synchronous/asynchronous communication as well mobility, size and power of system components.

Finally, any and all enumerations or lists found in the paper are not meant to be exhaustive but should be treated as examples. We recognize, as many before us have, that where policy is concerned, full enumeration of possibilities is impossible.

Organization: In the next section we motivate the need for two basic elements: group charter and group authority. Then, sections 3 and 4 set the stage for the framework by discussing the important variables and dimensions in admission control. The framework is presented in Section 5. The rest of the paper deals with group-based admission control. Sections 6 and 7 focus on voting-like admission methods and discuss several types of candidate signature schemes. Section 8 provides a brief overview of related work and Section 9 identifies some topics for future work.

2 Basic Elements

Our initial (and obvious) observation is that any peer group must have a well-defined procedure for admitting new members. Admission policies can be arbitrary and enumerating them is not useful. However, prospective group members must be able to get a clear idea of how to gain admission. This triggers the need for some form of an electronic document codifying admission rules. We refer to this document as a *group charter*. In subsequent section we will discuss the potential format of a group charter and its role in concrete admission procedures.

The simplest form of group charter is a Unix-style Access Control List (ACL). Most flavors of Unix include a group ACL in the "etc/group" file. This file, typically manipulated only by the superuser, lists all local groups and their respective membership. The Unix scenario is very restrictive since: 1) group ACL is a static entity, 2) membership in all groups is controlled by a single authority, and 3) group ACL is only meaningful within one Unix system or site. However, we note that the Unix approach was motivated by the need to control access to local resources such as files and printers.[1] Therefore, Unix-style groups are just a convenient abstraction that simplifies resource ACLs such as file read permissions. In fact, Unix groups are more akin to *roles* in Role-Based Access Control (RBAC) [32].

This is very different from the setting where group members *communicate* and the *communication* itself is the primary resource that must be protected. In our case, admission control can take many flavors and, while the explicit ACL-based approach is viable for some peer groups, other, more dynamic admission policies must be considered.

We argue that group charter, though necessary, is insufficient for effective peer group admission control. The missing element is the entity that can certify (or vouch for) group admission. This entity which we refer to as *Group Authority* or GAUTH for short, is authorized to issue Group Membership Certificates (GMCs). A GAUTH may be specific to a group or group-independent, i.e, it may serve a single or multiple groups. One natural GAUTH example is a Certification Authority (CA) or its delegate. Another example is the group itself, collectively, and yet another is the group founder. (For example, a group started by Alice and named appropriately – say, "Alice.friends" – might specify Alice as the GAUTH in the group charter.) However, as will be discussed later in the paper, the group itself is not a practical choice for a GAUTH.

Figure 1 shows three key elements of group admission. The topmost part reflects the creation of the group charter. For the sake of simplicity, we assume that the group charter is always signed by an external off-line authority, most likely a CA.

The second part of Figure 1 depicts the interaction between a prospective member and the group. Before attempting to join, a prospective member must, at the very least, know the *group name*. The next step

[1] Also, the entire user/account population is well-known in Unix.

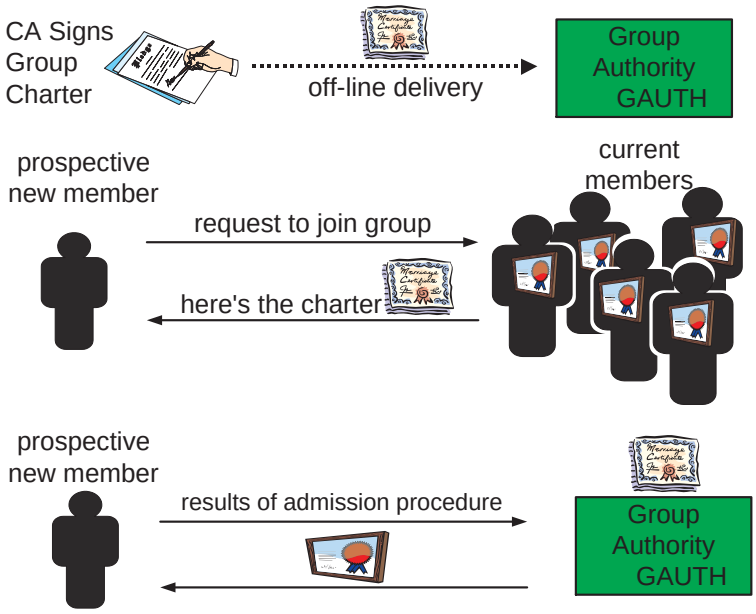


Fig. 1. Basic Framework

is obtaining the group charter. A natural place to obtain it is from the group itself. Another possibility (especially, if the group is fully asynchronous and no current member is on-line) is to obtain the group charter from a directory service that already serves as the distribution point for other certificate types, e.g., LDAP [39].

The last part of the figure illustrates the final interaction between the new group member and the GAUTH. Before approaching the GAUTH the new member is assumed to have satisfied all requirements for admission. The GAUTH's function is thus mechanical: it checks that admission requirements are indeed satisfied and, if so, issues a GMC.

Figure 1 omits the crucial stage of group admission – the group admission process itself. This is done on purpose since the admission process can vary widely among different groups. Much of the rest of this paper is dedicated precisely to the exploration of peer group admission processes.

3 Notation and Components

We now briefly present the notation and formats used in the remainder of the paper.

M_b^A - group member; the superscript refers to the group in question (e.g., "Poker.Players") and the subscript denotes the actual identity of the member (e.g., "Alice" or "Bob").

$Cert_b$ - public key identity certificate (PKC) issued by a recognized certification authority (CA). It binds a name (string specified in the subscript) to a public key. In the rest of this paper we assume X.509v3 certificates [1, 22].

GC^A - group charter of the group referred to in the superscript. As mentioned above, this is essentially an attribute certificate.

$\mathbf{GAUTH}^A$ - identity (Distinguished Name) of the group authority for the group referred to in the superscript.

GMC_b^A - group membership certificate for the group (superscript) issued to an entity (subscript) by the group authority GAUTH^A.

A group charter, GC^A, contains at least the following fields:

– GN - Group Name (A)

– GAUTH- Group Authority Name
– APT - Admission Policy Type (see below)
– IN - Name of the issuer (signer) of GC^A
– SIG - Signature (by issuer) of GC^A

In the above, names can be assumed to follow, for example, the X.500 Distinguished Name (DN) syntax.

The APT field reflects the admission policy type for the specified group. We identify the following types (keeping in mind that the list is not exhaustive):

– `APT_ACL`: explicit Access Control List (ACL) containing a set of names (or PKC serial numbers, or both) of entities allowed to join the group. A negative ACL (NACL), i.e., a list of entities not allowed to join, is very similar and can be easily accommodated as well as any combination of the two.
– `APT_GAUTH`: admission control at the discretion of the group authority. Here, the GAUTH can be the group founder, CA or a trusted third party (TTP).
– `APT_GROUP`: Access control by group members. This is further categorized into:
  • `APT_Group.Static` requires a fixed number (threshold) of sponsors for admission. (Whenever the number of current members falls below the threshold, special exception policy must be used.)
  • `APT_Group.Dynamic` requires a certain fraction (percentage) of current members sponsoring admission. (This policy type must also include the "bootstrap" rules for admitting at least the first member.)
  • `APT_Group.Hybrid` a blend of `APT_Group.Static` and `APT_Group.Dynamic`) models.

As mentioned earlier, a group membership certificate (GMC_B^A) for group A can be issued to a new member B by the group authority GAUTH^A only if B conforms to the group charter and can produce evidence to that effect.

We note that a GMC is not always necessary. If the APT field specifies `APT_ACL`, there is no need for GMCs, since any entity whose name is explicitly listed on the ACL can prove group membership by demonstrating knowledge (e.g., by signing a message) of a private key corresponding to a public key which is, in turn, bound to the said name in a regular PKC. Furthermore, a GMC is not always meant to be made public whenever proof of group membership is required. For example, group membership might be anonymous, in which case a member would only prove possession of a valid GMC instead of revealing it. (This is not difficult to achieve with advanced cryptographic techniques such as identity escrow [25] and group signatures [10].)

4 Dimensions of Group Access Control

We now turn to the discussion of important dimensions in peer group admission control. They are summarized in the rest of this section.

Membership Dynamics: Group membership dynamics can influence the basic design of group admission control mechanisms. We consider both static and dynamic groups. In a static group, the information about all prospective group members is known in advance. This information may include details such as names of all group members or blanket requirements for membership (e.g., anyone can be a member as long as they have an X.509 PKC issued by the University of California). In the former case, admission control can be based on ACLs, e.g., Kerberos-style access control [28] can be used. The latter case may be implemented via static group policy (which can be viewed as a type of an ACL). In a dynamic group setting, membership might be impossible to enumerate, either explicitly (e.g., via an ACL) or implicitly (e.g., via an attribute or a set thereof). We believe that the dynamic membership case is particularly interesting since it is an important trend among current peer group applications, such as Gnutella.

Membership Awareness and On-Line Presence: Some on-line peer group applications, notably those requiring reliable group communication, mandate constant awareness of group membership, e.g., Totem, Horus and Transis [2, 31, 3]. Others are asynchronous, such as mailing lists and Gnutella, and do not even announce membership changes. Since our admission control framework aims to cover most possible group settings, we do not make any assumption regarding membership awareness. In other words, an entity is only required to

know whether it is a member of a particular group and should be able to prove membership. For the same reason, we are also not concerned with the on-line presence of all group members. (This can be problematic if the number of current members needs to be known in order to perform admission according to a dynamic group admission policy (`APT_Group.Dynamic`) which requires sponsorship by a certain fraction of group members.)

Group Lifetime: The longevity of a group is an important factor in admission control. A solution appropriate for long-term group will be likely unsuitable for a short-term group. In the former, group membership may need to be revoked for the usual reasons, whereas, revocation is not nearly as important in a short-term group. Also, we note that a typical short-term group would not need codified admission control policy or mechanisms. (Especially considering that the initial signed group charter must be obtained from an off-line CA.) Spurious peer groups are therefore more likely to adopt some ad hoc admission policy. Consequently, from here on, we focus on medium- to long-term peer groups.

GAUTH Placement: Group admission decisions can be made internally, by one or more group members, or by an external entity. Assuming an external entity can be viewed as a violation of the *peer* nature of a group. However, for practical purposes – mostly because of the need to issue GMCs– we may need to require an external GAUTH. An off-line GAUTH is preferable since an on-line GAUTH cause the usual concerns with it being a single point of failure and a natural attack target.

GAUTH Composition: Admission decisions can be made by a single or multiple entities. When performed by a group founder, a TTP (e.g., Kerberos [28]), or a CA [16, 22], mechanisms tend to be uncomplicated since only a single party has to support a new member's admission. However, we also envisage group admission policies requiring, e.g., support of a certain number of group members. This can take on the form of voting where, for example, simple majority (or some other percentage) is required for admission. Alternatively, a fixed number of sponsors might be required. In either case, all group members collectively form the GAUTH. Mechanisms in support of such multi-party GAUTH admission are clearly more challenging and interesting.

5 Peer Group Admission Control

In this section, we introduce and discuss four models for peer group admission control. The first three require an outside GAUTH for all admission decisions while, in the fourth, the group members themselves play the GAUTH role.

5.1 Admission via Public ACL: `APT_ACL`

In this simplest scenario, potential group membership is enumerated *a priori*, i.e., all members are known at the time of group charter creation. The group charter becomes essentially a signed ACL and, as alluded to above, no explicit membership certificates are needed. There is also no need for a GAUTH since group admission is based on a public ACL contained in a (signed) *GC*.

When two (or more) group members communicate, they can simply sign all messages after exchanging their PKC s. Any message signed by an entity whose PKC is listed on the ACL is then deemed as emanating from a valid group member. (Of course, a message also has to be *intended* for the group, but that is a different matter altogether.) This approach is the most trivial as it requires neither an on-line TTP nor any real admission protocol. The main problem issue here is the inflexibility due to static membership.

5.2 Admission by GAUTH: `APT_GAUTH`

In this case, admission is performed by a GAUTH who ultimately issues a GMC to each incoming member. We stress that the admission process and the issuance of a GMC is performed by the same party, the GAUTH (who acts as both judge and jury). Naturally, the GAUTH must be trusted by all current and prospective members. To make the admission process efficient, the GAUTH has to be on-line (continuously present and available) and be resistant to hostile attacks. An on-line GAUTH is thus attractive for synchronous peer

groups. A special case of an on-line GAUTH is the group founder or an otherwise designated member. A variation on the theme is admission by any one of multiple GAUTHs ("OR" clause) listed in the group charter.[2]

Having an off-line GAUTH (e.g., co-located with a CA) would bring the usual benefits but would slow down the admission process. An off-line GAUTH is only appropriate for asynchronous groups.

5.3 Admission by Members: `APT_GROUP`

If the group charter stipulates admission by the group (`APT_GROUP`), things get more interesting. We distinguish between static and dynamic thresholds. (Hybrids are also possible but we do not discuss them here.) A static threshold is essentially a t-out-of-n model where t is fixed and n (current group size) varies over time. Whenever $n < t$, special fallback admission policy must be present in the group charter.

Figure2 illustrates the steps common to all schemes where admission is performed by the group.

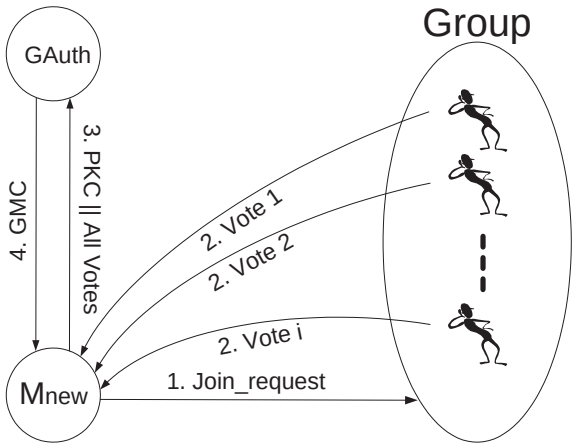


Fig. 2. Admission by Group Members

Step 1: Join Request
A prospective member M_{new} first sends a `join_request` to the group. This message is signed by M_{new} and contains, among other values, M_{new}'s certificate ($Cert_{new}$) and the target group name (say, "Geeks"). How this request is transported to the group is application-dependent. Note also that M_{new}'s certificate does not have to be an identity certificate; it could well be a group membership certificate for another group.

Step 2: Voting
Upon receipt of the `join_request`, a group member first extracts the sender's PKC and verifies the signature. If a voting member approves of admission it replies to M_{new} with a signed (and well-typed) message. The particulars of the signature scheme depend on the group charter (see below). Threshold signatures, group signatures, subgroup multisignatures as well as plain individual signatures are some of the possible techniques.

Step 3: GMC Request
Once enough votes are collected (according to the group charter GC^{Geeks}), M_n sends a signed `GMC_request` message to GAUTH^{Geeks}, which includes (at least) its PKC, group name, and the collection of votes.

[2] Yet another variation is a GAUTH represented by a fixed size subset of an explicitly listed set of members.

Step 4. GMC Issuance

Upon receiving the request, GAUTH^{Geeks} verifies the request (including the PKC and the signature), the individual votes contained therein and conformance to the group charter. Finally, if all checks succeed, GAUTH^{Geeks} issues a new GMC_{new}^{Geeks} to M_{new}

Armed with a new membership certificate M_{new} can act as a *bona fide* group member. To prove membership to another party (within or outside the group) M_{new} simply signs a message (challenge) to that effect.

An important issue has to do with the nature of the signature scheme that is used to prove membership, which is of course tied in with the nature of of a membership certificate. We discuss this in subsequent sections.

5.4 Group Acting As Its Own GAUTH?

It is worth exploring whether the group itself issue membership certificates. We argue that, even though group admission is conducted by group members themselves, a distinct group authority must issue group membership certificates.

If each group member votes to admit M_{new}, the latter can collate the necessary number of votes and use that as evidence of membership. However, if the number of required votes is large, this can become unscalable. A more important observation is that:

Each current member's vote must itself be accompanied by a proof of membership.

This is a classic example of the "chicken-and-egg" problem.

Cryptographic techniques are available that would allow a certain number of group members to collectively issue a (single) membership certificate. For example, robust threshold signatures can be used for that purpose [18]. However, most – if not all – such techniques are geared towards static groups. Any subgroup of at least t members (out of n total) can sign on behalf of a group while both t and n are fixed. A change of either (or both) t or n usually requires all members to interact with a trusted dealer. (There are some exceptions as in the case when n shrinks but remains greater than t.) We discuss this further in Section 6.

5.5 Summary

Of the three group admission approaches discussed thus far, the first (ACL-based) is by far the simplest. Its major drawback is the inflexibility. The second (GAUTH-based) is more flexible but introduces the reliance on a third party (external or internal) whose constant presence and security can become problematic. In addition, the need for a third party can be viewed as a violation of the *peer* nature of the group. In contrast, the last approach involves admission control by the members themselves which is certainly in the spirit of *peer* groups. On the other hand, mechanisms for group-based admission are more complicated, requiring multi-party and, possibly, multi-round, protocols. We believe that this is where the challenge lies and, therefore, focus on group-based admission in the remainder of the paper.

6 Admission by Voting

As discussed earlier, different flavors of voting may be used depending on the particular group admission policy reflected in the group charter:

Fixed : the minimum number of votes required for admission is constant throughout the group's lifetime. The problem with this policy is when the total number of group members is less than k. In this case, special admission rules are necessary.

Dynamic : the minimum number of votes is a fraction of the number of current group members. The main problem here is the need to securely and reliably determining the number of current group members. This is a harder problem than it seems since the members themselves cannot be relied upon to keep track of the current membership count. (Otherwise, we would need a group communication system providing strong membership semantics. This would restrict us to on-line, synchronous groups.)

Hybrid : many hybrid variations of Fixed and Dynamic policies are possible. One straightforward example is to use a fixed voting when the number of group members is small (less than some threshold t) and dynamic voting otherwise. It is just as viable to reverse this strategy. We note that in either case there remains the nagging issue of determining the current group size.

7 Sorting out Signature Schemes

Regardless of the admission policy type, voting can be realized with different digital signature techniques. There are many candidate schemes that vary in efficiency, security properties and other aspects. We now discuss their characteristics on the basis of which suitable signature schemes can be selected for specific peer group settings.

7.1 Plain Digital Signatures

Plain digital signature are a natural and default candidate for peer group access control. They can be used for all policy types. In the admission policy specifies `APT_ACL`, a prospective member presents its identity PKC along with a signature proving knowledge of the corresponding private key. If the said PKC is listed in the ACL, admission is self-evident and no group membership certificate is needed.

In case of `APT_GAUTH`, a GAUTH ultimately issues a GMC to the new member. As part of the issuance procedure, the member (or GAUTH or both) generate a signature key-pair which subsequently serves as this member's group-specific membership key. If the GAUTH does not refer to the member's name (or its PKC) in the GMC, the group member is anonymous to everyone except the GAUTH), i.e., its identity as a group member is divorced its "real" outside identity.

Using plain signatures with `APT_GROUP` is also very easy. In both static and dynamic cases, a group member simply signs an applicant's join request. An important advantage is that each current member can sign the join request **asynchronously**, i.e., the admission process does not require coordination among current members; the applicant can approach each member at its own leisure and gradually collect the required number of votes. The main drawback is the need to retain a linear number of votes before approaching the GAUTH. Also, a separate signature verification for each vote can amount to significant overhead.

7.2 Threshold Signatures

Another promising direction is to use a threshold signature scheme as the admission mechanism. Threshold cryptography was introduced by Desmedt and Frankel [15]. In a (k, n)) threshold scheme, a secret is split by a trusted dealer into n shares. Each share is given to a member and the secret can be reconstructed whenever at least k members pool their shares [34]. One typical application is the protection of the CA's private key. An attacker must compromise t members in order to produce valid signatures. Modern threshold signature schemes do not require reconstruction of the secret key; instead, function sharing is used to compute signatures.

Threshold schemes are only applicable to `APT_GROUP` admission policy since neither `APT_ACL` nor `APT_GAUTH` involve voting or collective participation by current group members. Also, threshold signatures are useful only for group admission; they are not a means for proving group membership.

Many flavors of threshold signatures have been proposed, e.g., verifiable [13], proactive [11] and robust [18]. However, we are mainly interested in the semantics and the architectural components of these schemes. In particular, the ability to change either (or both) k and n is crucial for the `APT_GROUP.Dynamic` policy type.

Fixed Threshold: We restrict our discussion to threshold RSA signatures since the other alternative (DSS) requires multiple rounds and coordination among signers. In the simplest threshold RSA scheme, each member has a secret share originally assigned by the dealer. When M_{new} requests to join, each available current member M_i (who favors admission) sends a partial signature to M_{new}. Unfortunately, M_{new} is unable to compose the signature even when it collects k votes since it cannot compute Lagrange coefficients. One solution is for the new member to forward the set of partial signatures to some trusted dealer who would

construct and return the threshold signature. If the trusted dealer is also the GAUTH, this is a relatively efficient approach since the membership certificate can be issued simultaneously. (If the GAUTH is a distinct entity, more communication rounds would be necessary.)

An alternative is to use Frankel et. al.'s method [17] where all members compute partial votes, including Lagrange coefficients. The new member only needs to multiply partial votes to obtain the whole signature. A problem with this approach is that the group member needs to know who will take part in voting. In other words, this would require two extra communication rounds.

Dynamic Threshold: There have been a few results in threshold signatures where the threshold is dynamic. One such technique was proposed by Frankel et. al. in [17]. It uses proactive cryptography to dynamically adjust the threshold (t). This scheme can be modified to adhere to the general protocol shown in Figure 2. However, we recall that signing needs multiple communication rounds and simultaneous on-line presence of all signers. (Incidentally, proactivity also requires periodic re-freshing which further intensifies the on-line presence requirement.)

Some promising recent results show that it is possible to avoid the dealer (most of the time) by distributing its role among the members themselves. Kong et. al. [26] proposed a fully distributed (k, n) threshold signature scheme with a fixed k. During initialization, a trusted dealer is involved in generating an RSA modulus n and a secret function $f(x)$ where $f(ID)$ is distributed among each of the k members. (Exactly how these first k members are admitted is unclear.) At the admission stage, when a candidate requests to join, each member signs a partial certificate using its Lagrange coefficient and a secret share. The candidate then computes its new GMC and obtains its secret share.

This scheme can be viewed as an ideal solution for peer group access control. mainly because it offers minimal TTP involvement and the GAUTH is represented by the group itself. However, certificate issuance is still an on-line, coordinated multi-round protocol; also, the certificate issuers cannot be traced.

7.3 Accountable Subgroup Multisignatures

Another signature type potentially useful for peer group admission control is called "Accountable Subgroup Multisignatures" (ASM) originally proposed by Ohta et. al. [29]. An ASM scheme enables any subgroup, S, of a given group, G, of potential signers, to sign efficiently in a way that the signature provably reveals the identities of all individual signers to any verifier.

Ohta et. al. take advantage of the homomorphic property of Schnorr signatures [33] to construct an efficient ASM scheme. We briefly summarize the protocol below. (A detailed description can be found in [29].)

To sign a message (a join request in our context), each current group member sends a partial commitment to the verifier. The verifier multiplies k commitments to obtain the joint commitment which is then sent – along with a challenge – to the k members. The joint commitment contains the names of all participating members. The members compute and return partial signatures back to the verifier (prospective member). Finally, the verifier constructs a complete signature by summing up the partial signatures.

The verification phase in this scheme requires only two modular exponentiation and k modular multiplication, since an ASM is effectively the same as a regular Schnorr signature. This is very efficient since, otherwise, k verifications would be performed. Another notable feature of ASMs is the full accountability of signers. Unlike threshold signatures, no dealer is assumed. However, a GAUTH is still needed to issue GMCs since otherwise, certificates of all signers (sponsors) must be presented in order to prove membership. This can become very complicated in a dynamic group.

7.4 Group Signatures

In a group signature scheme (e.g., [12, 10]) all group members are peers and any member can sign on behalf of the group in an anonymous and unlinkable manner. Unlinkability is generally a desirable feature, but it implies that accountability is hard to achieve. It is, however, not impossible since group signature schemes include a provision that allows a designated entity (Group Manager) to open a signature and identify the actual signer. The same Group Manager is also the entity issuing membership certificates. (Except that in this setting, certificates are never revealed as part of signing.)

The only glitch with group signatures is that, as specified, they do not provide a means to distinguish among signers. More precisely, it computationally hard to decide whether two group signatures are produced by the same, or two distinct, signers. Fortunately, simple add-on techniques have been proposed to address this issue. For example Ateniese, et. al. [7] illustrate a simple extension scheme for group signatures to obtain sub-group signatures (where the number of distinct signers is evident).

7.5 Comparison

	Key Features					
Signature Type	Prove membership	On-line presence	Membership awareness	Accountability	Anonymity	Unlinkability
Plain	YES	NO	NO	YES	YES	NO
ASM-s	YES	YES	YES	YES	YES	NO
Threshold fixed	NO	YES	NO	NO	YES	YES
Threshold dynamic	NO	YES	YES	NO	YES	YES
Group	YES	NO	NO	NO*	YES	YES*

Table 1. Admission Control Signature Mechanisms: Feature Summary

The signature schemes we discussed above offer very different alternatives for peer group admission control. Table 1 summarizes their key features. (Some information in the table is redundant, e.g., the accountability and uninkability columns are each other's complements. We list them separately since – depending on the setting – each can be a desirable or an undesirable feature.)

Plain, ASM and group signatures can be also used for proving membership once the admission is over, whereas, threshold signatures cannot. Plain and group signatures are the most general in that neither on-line presence of all signers nor membership awareness is necessary. This is important for asynchronous, off-line groups such as mailing lists and file sharing communities.

Only conventional digital signatures and ASMs directly identify the sponsors/signers. Signers' accountability is impossible with threshold signatures and possible, but awkward, in group signatures. (We remark that accountability is not always desired.) The latter two offer inherently different semantics: threshold schemes require a certain number of signers sign a message jointly, while group signatures allow any member to sign anonymously and untraceably on behalf of the group.

ASMs and plain signatures cannot be made unlinkable in contrast to threshold and group signatures where unlinkability is built in. All schemes can offer anonymity (actually, pseudonymity) provided that the membership certificate is not tied in directly with the member's PKC. (The member's identity within the group does not have to relate to its identity outside the group.)

Plain digital signatures rely on each signer having its own key-pair which is independently generated. The same independence of key generation applies to ASM schemes. In contrast, threshold and group signature schemes entail a complicated setup phase and a non-trivial join procedure. Moreover, the complex setup does not eliminate the need for having specific key material for signing messages within the group.

8 Related Work

The Antigone [27] project is the closest related work. Antigone includes a flexible framework for secure group communication and utilizes a centralized admission approach geared primarily towards secure multicast scenarios. Antigone offers flexible mechanisms for defining policies about membership, application messages and other aspects.

In Antigone, member admission is mediated by a Session Leader (SL) which interacts with the TTP (that operates on-line) in order to admit a new member. The TTP shares a symmetric key both with the SL and every potential new member. (The TTP acts like a Kerberos AS/TGS). Everyone is expected to know in advance the identity of the SL.

There have been other efforts to develop standard frameworks for creating peer-to-peer applications, for example, JXTA [36] (an open-source project initiated by SUN) and Peer-to-Peer Trusted Library (PtPTL),

also an open-source project sponsored by INTEL [37]. JXTA uses SSL/TLS as its security mechanism while PtPTL supports a wide variety of options. There is also an active working group within the IETF (P2PWG) [38] created with the charter to facilitate and accelerate the advancement of common mechanisms peer-to-peer computing. One of the documents produced by P2PWG is an internet draft addressing the security requirements for peer-to-peer applications; it identifies authorization as one of the major issues.

Some of the mechanisms discussed in this paper are akin to limited forms of voting. Electronic voting schemes have been extensively studied starting with the seminal work of Benaloh [8]. Most approaches are based on mix-nets, homomorphic encryption [14] or blind signatures [30]. Usually, voting schemes must satisfy many different requirements, e.g., privacy, anonymity, unlinkability, un-coerceability and, more recently, receipt-freeness [21]. However, the framework proposed in this paper, does not deal with secret-ballot election. Instead we focused on mechanisms for registering consensus about a particular event type (new member admission).

9 Conclusions

In this paper we (hopefully) motivated the importance admission control in dynamic peer groups. This work represents an initial attempt to construct an admission control framework suitable for different flavors of peer groups and match them with appropriate cryptographic techniques and protocols. We examined various dimensions of admission control, discussed several cryptographic techniques and assessed their applicability.

There remain many items for future work. In particular, revocation of group membership is left untouched in this paper. While relatively well-understood in regular PKI settings, revocation of group membership has not been adequately investigated in more complicated settings such as threshold signatures. In group signatures, a few recent results yield rather inefficient revocation methods, e.g., [35] and [9].

Some of the assertions and observations (especially about the efficiency of various signatures schemes) made in this paper are not clear-cut and cannot be proven without some experimental results. To this end, we need to develop a toolkit in support of major peer group admission policy types (ACL-based, GAUTH-based and group-based) and measure their performance in realistic peer group settings.

In group environments where secure any-to-any communication among all members is needed, group admission needs to be tightly integrated with group key management. Only then can the total overhead of joining a peer group be measured.

References

1. C. Adams and S. Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. RFC 2510, IETF, Mar. 1999.
2. D. Agarwal, L. E. Moser, P. M. Melliar-Smith, and R. K. Budhia. The totem multiple-ring ordering and topology maintenance protocol. *ACM Transactions on Computer Systems*, 16(2):93–132, May 1998.
3. Y. Amir, D. Dolev, S. Kramer, and D. Malki. Transis: A communication sub-system for high availability. *Digest of Papers, The 22nd International Symposium on FaultTolerant Computing Systems*, pages 76–84, 1992.
4. Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, J. Stanton, and G. Tsudik. Exploring robustness in group key agreement. In *21th IEEE International Conference on Distributed Computing Systems*, pages 399–408. IEEE Computer Society Press, April 2001.
5. Anonymized. Anonymized. *Anonymized*, 200X.
6. Anonymized. Anonymized. In *Anonymized*, 200X.
7. G. Ateniese and G. Tsudik. Some Open Issues and New Directions in Group Signatures. In *Third International Financial Cryptography Conference*, 1999.
8. J. Benaloh. Veriable Secret-Ballot Elections, Yale University PhD thesis. YALEU/DCS/TR-561, 1987.
9. E. Bresson and J. Stern. Efficient revocation in group signatures. In *Public Key Cryptography (PKC) Conference*, pages 190–206, 2001.
10. J. Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology—EuroCrypt'97*, pages 465–479, May 1997.
11. R. Canetti and A. Herzberg. Maintaining security in the presence of transient faults. In Y. G. Desmedt, editor, *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 425–438. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1994.

12. D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT'91*, May 1991.
13. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *26th Symposium on Foundations of Computer Science (FOCS)*, pages 383–395. IEEE Computer Society Press, 1985.
14. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Theory and Application of Cryptographic Techniques*, pages 103–118, 1997.
15. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, number 435 in Lecture Notes in Computer Science, pages 307–315, Santa Barbara, CA, USA, 1990. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany.
16. S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization. RFC 3281, IETF, Apr. 2002.
17. Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung. Optimal-resilience proactive public-key cryptosystems. In *38th Symposium on Foundations of Computer Science (FOCS)*, pages 384–393. IEEE Computer Society Press, 1997.
18. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold dss signatures. In *Advances in Cryptology—Eurocrypt'96*, May 1996.
19. Global Positioning System, available from `http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html`.
20. Gnutella.com, available from `http://www.gnutella.com`.
21. M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology—EUROCRYPT'00*, pages 539–556, May 2000.
22. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, IETF, Apr. 2002.
23. IETF The Multicast Security (MSEC) Working Group, available from `http://www.securemulticast.org/msec-index.htm`.
24. IRTF The Group Security (GSEC) Research Group, available from `http://www.securemulticast.org/gsec-index.htm`.
25. J. Kilian and E. Petrank. Identity escrow. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO '98*, number 1462 in Lecture Notes in Computer Science, pages 169–185. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1998.
26. J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for MANET. In *IEEE 9th International Conference on Network Protocols (ICNP)*. IEEE Press, 2001.
27. P. McDaniel, A. Prakash, and P. Honeyman. Antigone: A flexible framework for secure group communication. In *8th USENIX Security Symposium*, pages 99–114, Washington, D.C., USA, Aug. 1999. USENIX.
28. B. C. Neuman and T. Y. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, Sept. 1994.
29. K. Ohta, S. Micali, and L. Reyzin. Accountable-subgroup multisignatures. In *8th ACM Conference on Computer and Communications Security*, pages 245–254. ACM Press, November 2001.
30. T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols Workshop*, pages 25–35, 1997.
31. R. V. Renesse, K. Birman, and S. Maffeis. Horus: A flexible group communication system. *Communications of the ACM*, 39(4):76–83, April 1996.
32. Role Based Access Control, available from `http://csrc.nist.gov/rbac/`.
33. C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
34. A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, Nov. 1979.
35. D. Song. Practical forward secure group signature schemes, Nov 2001.
36. The JXTA project, available from `http://www.jxta.org/`.
37. The Peer-to-Peer Trusted Library project, available from `http://sourceforge.net/projects/ptptl/`.
38. The Peer-to-Peer Working Group, available from `http://www.peer-to-peerwg.org/`.
39. M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). RFC 2251, IETF, Dec. 1997.