Introducing Secure Peergroups in SP^{2}A

Michele Amoretti, Matteo Bisi, Francesco Zanichelli, Gianni Conte
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze, 181A, I-43100, Parma, Italy
{amoretti, mbisi, zanichelli, conte}@ce.unipr.it

Abstract

Service-oriented Grids are aiming at new applications beyond traditional resource-oriented scientific computing. We argue that Community Support is emerging as the real killer application. Service-oriented systems provide a flexible solution for work groups which belongs to University Campuses, Research Labs, Enterprises, Finance environments.

We are building the Service-oriented P2P Architecture (SP^{2}A), which is the result of the positive convergence between Grid and Peer-to-Peer computing. P2P design allows to overcome the limitations of available mechanisms for publishing and discovering Grid Services, which are essentially based on centralized directories, thus raising robustness, scalability and performance concerns.

SP^{2}A is constituted by distinct communities of peers organized for specific purposes. In this paper we illustrate the group security policies of SP^{2}A, which define the mechanisms for peer authentication, peergroup admission control, authorization and transport security. Prototype implementation details are also exposed.

1 Introduction

The Peer-to-Peer (P2P) approach can help bringing service-oriented Grid technologies to new dynamic application environments and beyond its traditional domain of resource-oriented scientific computing. For example, higher levels of flexibility and scalability for service publishing and discovery can be obtained by replacing or complementing centralized directories (*e.g.* UDDI registries) with P2P-oriented distributed indexes.

Given the primary issues of trust and security, including but not limited to the usual need for mutual authentication of servers and clients, a P2P-based service-oriented solution needs to cope with challenging requirements. Unlike a centralized registry containing service descriptions of service providers, which have been approved and thus endorsed by the registry maintainers (trusted themselves because of their commercial power), a P2P-based decentralized management of Grid Service advertisements calls for new tradeoffs between the liberal but risky possibility for providers to expose self-promoted services and the inflexible security available in Grid systems.

Our *Service-oriented Peer-to-Peer Architecture (SP^{2}A)* [9] is essentially a network of peer Service Host Environments built upon the Open Grid Service Architecture (OGSA) [15], and the framework defined by Project JXTA, Sun Microsystems' open P2P initiative [2]. SP^{2}A takes advantage of the strengths of these two technologies at different levels, by relying on JXTA P2P network as a decentralized carrier of light/simple service advertisements and on Grid Service query and invocation mechanisms which exploit robust and secure technologies. This approach allows to cope with the requirements of applications with a large number of participants dynamically connecting to the system, and provides high levels of scalability, decentralization and interoperability.

Regarding security, we have recently started working to introduce interest-based secure peergrouping to the SP^{2}A prototype. In this paper we initially discuss the motivations underneath this choice and outline some relevant issues. Then we illustrate the envisioned group security policies of SP^{2}A and define several mechanisms for peer authentication, peergroup admission control, authorization and transport security. Finally, we provide some details about the prototypal implementation we are currently testing in our University Campus.

The paper is organized as follows. Section 2 illustrates the benefits of peergrouping, in general and in particular for our service-oriented architecture. How peergroups can be secured is explained in section 3. In section 4 we illustrate the prototype under development, providing some details about the implementation of SP^{2}A authentication mechanisms, and the framework on which we are building SP^{2}A membership policies. Section 5 discusses related work on group security policies. Finally, an outline of open issues concludes the paper.

2 SP^{2}A Peer Communities

In a global knowledge-based society, communities play a pivotal role and re-shape the process of learning and sharing knowledge in and among organizations. In a P2P network space, peergroups represent communities of peers organized for specific purposes. It is quite obvious that, by forming a group, the nodes can lower the number of queries they receive and increase the efficiency of the queries they generate, but this is not the only reason of peergrouping. The creation of (disjoint or overlapping) subspaces can be also motivated by the need to create scoping environments which restrict the propagation of query messages, thus improving the performance of discovery algorithms. Moreover, content exchange and service interaction could require the creation of secure domains. Finally, subgrouping can simplify traffic inspection and tracing, thus enabling the creation of monitoring environments.

Two approaches can be adopted: self-organization or user-driven subgrouping. In the first case, peers can be categorized according to the type and quality of service they provide, resource owned, geographical location, and a number of techniques can be adopted to automatically group peers into similarity groups [10, 18]. In the other case, when a peer joins the network, it tries to discover the group which may have interest in the capabilities/services provided by the new peer. As for discovery of an existing group, the creation of a new one depends on the user's interests. User-driven subgrouping is the solution currently adopted in SP^{2}A, even if we plan to study the applicability of the self-organization approach.

In the context of university campuses the ever increasing joint availability of portable devices and wireless technologies and their forthcoming functional integration are changing both the way in which students attend their education in the university and the way the institution and the professors communicate with the students as well as the way in which students communicate to each other. Our campus has been chosen for these reasons as the natural testbed for SP^{2}A peergroups, which can be created only by recognized and "high-rank" peers (*e.g.* teachers, research group heads, lab responsibles), while access is granted to other peers (students, researchers, technical staff, guests) with various roles. In the academic domain under consideration, a peergroup can be a cluster of peers which share similar interests and competencies, exposed as services. This type of peergroup is known as a *Virtual Knowledge Community (VKC)* [17].

It is quite evident that providing security to peergroups is rather a difficult goal, because interactions are not just user-to-service, but also service-to-service on behalf of the peers, thus requiring delegation of rights from peers to services and dynamical instantiation of services. Moreover, implementation must be broadly available and applicable, *i.e*. standard, well-tested, well-understood protocols are needed. Finally, a number of different policies from sites, VOs, peers need to be combined. In the following we illustrate the possible solutions we evaluated, and justify our choices for SP^{2}A. We emphasize that security in SP^{2}A is addressed both at the overlay P2P network level (JXTA) and at the service level (OGSA). We focus here on the security policies adopted at the overlay P2P network level, which exploit mechanisms for *peer authentication*, *admission control and authorization*, *transport security*. Secure service invocation issues are out of the scope of this paper.

3 Group Security Policies

A virtual organization policy must ensure that no participant uses an unfair share of the resources. Thus, group membership and service access generally require policies for:

- key management
- authentication
- admission control
- authorization
- transport security

Those policies can rely on two orthogonal approaches for group security: the reputation scheme, and the trust negotiation scheme [26]. In a reputation system, a peer makes decisions on its own experience and other peers' recommendations. Even though some form of persistent node identification is required, reputation systems are not considered suitable for critical tasks. This weakness is addressed by the other scheme, which builds trust by exchanging digitally signed certificates.

On the other hand, complicated security mechanisms can adversely affect performance. For this reason the ability to dynamically control security levels based on information known at runtime is a desirable feature.

In general, prospective group members must be able to know which prerequisite conditions they need to satisfy in order to gain admission to a group. This can be expressed by means of an electronic document, the *Group Charter (GC)*, which codifies the membership policies.

In the case of the proposed security policies for SP^{2}A, peers in the main peergroup, *i.e.* the NetPeerGroup (NPG), are only allowed to look for peergroup advertisements (each one containing the relevant GC), and for a *Certification Authority (CA)* peer issueing a signed certificate, which is necessary for authenticating the peer so that it can attempt

gaining admission to secure groups. The interaction between peer and CA requires a secure channel to communicate (encrypting messages). In the following we describe how these issues have been addressed by choosing the appropriate policies. External CAs may be present, providing offline certification for CA peers and possibly for peers. In the following we consider peers which are not provided with offline-signed certificates.

Figure 1 illustrates a typical SP^{2}A configuration. SP^{2}A is an unstructured supernode network, where simple peers can become supernodes dynamically, and vice versa. A node can be member of different groups, acting as a supernode for some of them. In the proposed example, we consider statically configured nodes and supernodes, which are all members of the NPG, and, except for CA peers, they are also members of (maybe overlapping) NPG's subgroups.

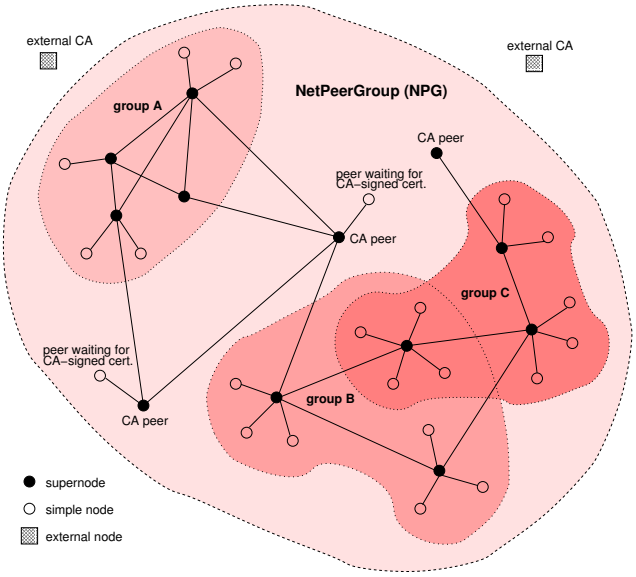

Figure 1. SP^{2}A overlay network: the NetPeerGroup (NPG) and its subgroups. CA peers and external CAs are also illustrated.

3.1 Peer Authentication

The peergroup creator should be able to choose a membership policy in a trust negotiation spectrum that has self-signed certificates near one endpoint, and certificates signed by a trusted third party [11] near the other one. SP^{2}A currently supports both the extreme solutions. Self-signed certificates are used to enter the main peergroup and to create secure channels (encrypted), in order to obtain certificates by trusted third parties.

Self-signed certificates are well-known to be left open to the man-in-the-middle attack. In a more dependable policy, a certificate binds each peer with its public key and personal description. Obviously, each peer's identity must be unique across the universe of peers. The certificates are assumed to be created by some trusted Certification Authority (CA), and have the following characteristics [25]:

- any peer with access to the public key of the CA can verify the peer public key that was certified
- no party other than the CA can modify the certificate without being detected

At a minimum, certificates contain the subject's name, the subject's public key information, the issuer name and the digital signature of the issuer. They may also contain a serial number, a period of validity, the issuer unique identifier and other extensions. An important standard for certificates is $X.509$ [3], which is used in a variety of contexts. For example, the X.509 certificate format is used in electronic mail security (S/MIME [5]), network layer security (IPSec [16]), transport layer security (SSL/TLS [24] and SET [14]).

Having identified the peer, the CA signs the peer's certificate with its own private key. If the corresponding public key is known, then any peer can verify that a certificate signed by the CA is valid. In a large community of peers, it may not be practical for all peers to subscribe to the same CA. In general, since a single CA means having a single point of failure, it is better to have multiple CAs, each one securely (with respect to integrity and authenticity) providing its public key to some fraction of the peers. Moreover, CAs need to be hierarchically organized in a *Public Key Infrastructure (PKI)* where trust propagates from the root CA to leaf CAs. In the SP^{2}A main group (of which all peers are member) there are stable and recognized CA peers (peers which offer a CA service) which issue X509.V3 certificates.

3.2 Admission Control and Authorization

Security mechanisms related to the evenly important aspect of admission control [21] can be categorized as ACL-based, GAuth-based and threshold-based. Using a static *Access Control List (ACL)* is the simplest approach, but it lacks for scalability. The threshold-based solution, in which all members of the group participate in the decision of admitting/rejecting a new peer, is interesting but difficult to implement (to be an effective solution, it should use a dynamic threshold). In an intermediate solution, the *Group Authority (GAuth)* is the entity which checks if all the admission requirements are satisfied by the prospective member. The GAuth can be a CA (internal or external), or some elected members of the group (distributed GAuth), or the group founder.

The latter is the approach we have chosen for SP^{2}A, whose peergroup members can have different ranks, corresponding to the actions they are allowed to perform within the group. Only peers with the highest rank, *i.e.* admins, can be members of the GAuth and provide *Group Membership Certificates (GMCs)*, which state membership and define the capabilities of each peer on behalf of the group. A partial list of ranks is:

- admin - the peer is a member trusted by the group founder, or it is the group founder itself; the actions it is allowed to perform are: service sharing/discovery, group monitoring, voting for changing member ranks;
- newbie - the peer is a new member; it only can search for an admin peer, to ask for a promotion;
- searcher - the peer is allowed to search for services and to interact with them;
- publisher - the peer can search for services but also publish its own services in the peergroup.

A newbie can become a searcher if the admin verifies its conformance to the GC (*e.g.* accepts its certificate). Becoming a publisher is more difficult because the peer must have a good reputation (high rating) and the agreement of more than one admin. These conditions, and the approval of the group founder, are required also for the promotion to admin rank.

Malicious peers could be interested in joining a peergroup not to use its services, but only to generate useless traffic (*i.e.* discovery messages, etc.) and decrease the efficiency of the subnetwork. *Client Puzzles* [20] represent an interesting countermeasure. When a service providing peer comes under attack, it distributes small cryptographic puzzles to its requesters. Refusing to either solve or bad solutions yields a refusal of the connection attempt. Moreover, peers which have been identified as malicious should be lowered in rank and blacklisted so they are ignored even by admin peers. Also peers which have a negative rating, *e.g.* because the services they offered were unsatisfactory, should be excluded from the group.

3.3 Transport Security

SP^{2}A secure communications are based on *Transport Layer Security (TLS)* [24], and RSA 1024bit as default cipher suite. When a secure pipe (*i.e.* virtual channel) is created, and the associated peer endpoints are resolved, a virtual TLS transport is instantiated. All data moved through Secure Pipes are then multiplexed over this single instance of a virtual TLS transport. The transport is bi-directionally secured end-to-end with TLS, independently of the underlying physical transports. The adopted TLS implementation minimizes the needed network resources by amortizing one TLS handshake over multiple data pipes and making conservative use of the bandwidth on the physical layer. Because the TLS virtual transport is bi-directional, both client and server authentication is required because peers may be both clients (sending data) and servers (receiving data). Peers must possess, at least, the X509.V3 self-signed certificate of any peer with whom they wish to communicate securely.

4 SP^{2}A Prototype Enhancements Supporting Secure Peergroups

For the development of SP^{2}A peers we are exploiting the following tools: the Globus Toolkit (GT) [1] and JXTA Java binding. GT is an open source implementation of the Open Grid Services Architecture (OGSA) [15]. JXTA Java Binding supports the implementation of P2P features, such as connectivity, peergrouping and service sharing/discovery.

GT offers the Grid Security Infrastructure (GSI) which provides support for *Single Sign-On (SSO)* [4] to access Grid Services. SSO is the mechanism whereby a single action of user authentication and authorization can permit a user to access all systems where he has access permission, without the need to enter multiple passwords. On the other hand, JXTA provides X509.V3 and TLS support and a framework for peergroup membership management. By the integration and extension of GSI and JXTA security APIs, we are enhancing the SP^{2}A prototype with secure peergroups, confidential communication between peers and secure Grid Service access. In the following we firstly describe how peers can be configured, and then the implementation of CA services and group admission control.

4.1 Peer Configuration

At boot time, every peer inizializes all basic peergroup services and protocols (see table 1) and creates a *PlatformConfig* advertisement. This advertisement also contains some peer's security informations, such as the *root certificate* (an X509.V3 compliant self-signed certificate) which binds the peer with a public key, and the corresponding private key stored in encrypted format.

To bridge to existing common membership and access technologies, such as PKI [3] and Kerberos [12], JXTA introduces the concept of MembershipService, which should allow the peer to establish an identity within a peergroup. Each MembershipService implementation is responsible for its own protocol definition. JXTA's implementation of PKI is named *Personal Security Environment (PSE)*, and since JXTA v2.3 the default choice for the NPG is the PSEMembershipService. This service inizializes also the *PSE KeyStore*, an object that acts as a secure datastore for peer certificates and keys. In order to use the Tl-

Table 1. Protocols and services supported by the peer, in its default configuration.

PROTOCOLS	TcpTransport ServletHttpTransport EndpointRouter TlsTransport CbJxTransport RelayTransport
SERVICES	EndpointService ResolverService PSEMembershipService AlwaysAccessService DiscoveryService RendezVousService PeerInfoService

sTransport, the PSEMembershipService instantiation is necessary.

The peer can create subgroups with extra capabilities such as customized protocols and specific security policies, by using the NetPeerGroup (NPG) as a template. The SP^{2}A peer can be member of the NPG and a fixed number of NPG subgroups (created by the peer or discovered in the NPG). Thus, the group tree has two levels, and no further subgrouping is allowed.

Like every other JXTA overlay network resource, peergroups are represented by XML documents, named advertisements. Peergroup advertisements have the following schema:

```
<xs:element name="PGA" type="jxta:PGA"/>
<xs:complexType name="PGA">
  <xs:sequence>
    <xs:element name="GID" type="jxta:JXTAID"/>
    <xs:element name="MSID" type="jxta:JXTAID"/>
    <xs:element name="Name" type="xs:string" .. />
    <xs:element name="Desc" type="xs:anyType" .. />
    <xs:element name="Svc" type="jxta:serviceParam" .. />
  </RootCert>
</Parm>
...
```

The peergroup ID (GID) is the unique identifier of the group. The peergroup Name can be unique if it is obtained from a centralized naming service which guarantee name uniqueness (in a Campus environment, this should be feasible). The Desc field is very important because the peergroup creator fills it with the Group Charter, which defines the group admission requirements. Finally, any number of Svc elements may exist, each of them describing the set of basic network services which are supported by the peergroup (not the OGSA services, anyway). These services are composed of a collection of instances running on the peergroup members, and are not affected by single peer failures.

4.2 Certification Authority Peers

Consider a peer which connects to the SP^{2}A network, thus becoming a member of the NetPeerGroup (NPG), finds a suitable (for the peer's interests) peergroup and decides to become member of that peergroup. All secure groups require that any prospective peer must own a CA-signed certificate containing its personal attributes, *e.g.* its Faculty and its Title (X509.V3 certificates have a number of extensions for the subject's attributes).

While some peers could have obtained in advance their CA-signed certificate by means of offline interactions with external CAs, there is the need to provide others with certificate issued by special CA peers belonging to the SP^{2}A network. The identity of requesting peers can be automatically verified in a CA peer by obtaining username/password credentials and validating them against a *Registration Authority (RA)*. The CA peers belong to a hierarchical structure of CAs, where trust propagates from the root CA to the leaf CAs. In an academic context, each Faculty or Department could have its CA trusted by the root CA of the University. A peer can ask for a certificate to any leaf CA peer, assuming that they all connect to the same centralized University RA to retrieve peer personal information, which are necessary to create their certificates. In a different scenario, each Faculty or Department has its private RA, and the prospecting peer must discover, in the NPG, the CA which should be able to create a certificate with all the peer's information.

We have developed a CA peer prototype (which is a SP^{2}A supernode), and we plan to use it to set up a number of Faculty leaf CAs. The peer-CA interactions, in SP^{2}A, can be summarized as follows:

1. the peer searches and identifies the CA;
2. the peer and the CA exchange their certificates (self-signed for the peer, root CA-signed for the CA);
3. the peer and the CA create a secure pipe (TLS-based) for the transmission of peer's requests and the acquisition of the CA-signed certificate.

We have also implemented the PSEManager, which provides peers and CAs with a collection of utility functions for managing the PSE KeyStore content.

The sequence diagram in figure 2 provides some details about the operations performed by the peer, which sends the *Certificate Signing Request (CSR)*, and by the CA, which identifies the peer, creates the signed certificate and sends it back to the peer. The CA is multithreaded, with the main thread (RdvCAMainThread) waiting for connections from peers, and as many service threads (RdvCAServiceThread) as the number of connected peers. CA service threads and peers create a PSEManager instance apiece.

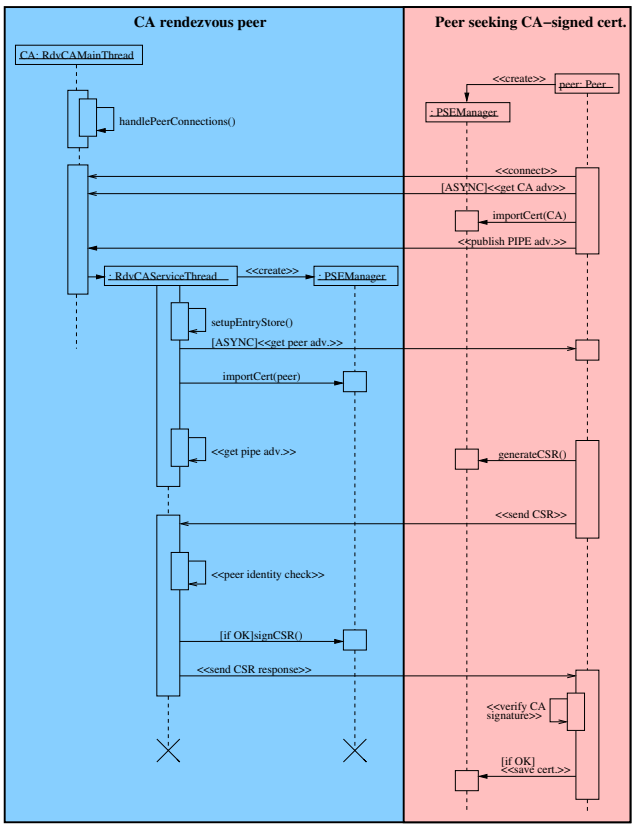

Figure 2. Interaction between CA and peer.

Each CA service thread imports the peer self-signed certificate into its own PSE KeyStore. Then it retrieves, in the apposite CA store, the pipe advertisement previously published by the peer. Both pipe advertisement and peer certificate are necessary to instantiate the secure pipe, on which the CA waits for the CSR from the peer. Once received the CSR, the CA checks the peer's identity on the RA (*e.g.* performing an interrogation on one of the LDAP servers which have access to personal data of University students, researchers, professors, and technical staff). A negative result produces the rejection of the CSR, the closing of the connection with the peer and the termination of the CA service thread. In case of positive result, a new certificate is created, built on the information provided by the self-signed certificate of the peer (which contains its public key) and by the RA, and signed with the CA private key. Finally, the CA service thread sends the certificate to the peer.

The peer, on its behalf, imports the certificate of the CA. Then, using the pipe advertisement previously published to the CA and the certificate of the CA, the peer is able to instantiate a secure pipe, on which it sends the CSR to the CA, and then waits for the CSR response. If the response is affirmative, the peer checks if the received certificate has been signed by the CA to which it was connected. For this purpose, it uses the public key which was acquired along with the CA certificate, which is signed by the root CA. If the the result of the CA signature verification is positive, the certificate is saved in the PSE KeyStore of the peer, replacing its self-signed certificate. At this point, the peer has new credentials (signed by a trusted third party) and it can try to join a secure peergroup.

4.3 Applying Membership Policies

When a peer tries to join a peergroup, being this the NPG or a subgroup, the membership policy of the peergroup establishes a temporary identity for the peer, for the sole purpose of allowing the peer to establish its identity by interacting with the membership policy (this may involve username/password insertion, exchange of public keys, or other mechanisms). The group admission framework provided by JXTA is constituted by the MembershipService, Authenticator, AuthenticationCredential and AccessService classes. The interaction of their instances is illustrated by the collaboration diagram in figure 3 and explained in the following.

1. An AuthenticationCredential is used by the MembershipService as the basis for applications for peergroup membership. The AuthenticationCredential provides two important pieces of information: the authetication method being requested, and the identity information which will be provided to that authentication method.

2. An Authenticator is returned by the apply() method of the MembershipService of a peergroup. The mechanism for completing the Authenticator object is unique for each authentication method. The isReadyForJoin() operation notifies that the peer has completed the authenticator correctly.

3. Next step is entering the peergroup, which is made through a join() invocation on the MembershipService. Thus, a new Credential for the new identity will be available to the peer.

4. The AccessService provides a mechanism to define which operations are permitted and which are not, for different kinds of peers. A PrivilegedOperation identifies an operation whose usage is restricted. PrivilegedOperations must be invoked through the doAccessCheck() method of the AccessService, passing the requestor's credential as parameter.

SP^{2}A secure groups have specific implementations of the above classes. In particular, the SecureGroupAuthenticator is able to compare the requirements defined in the

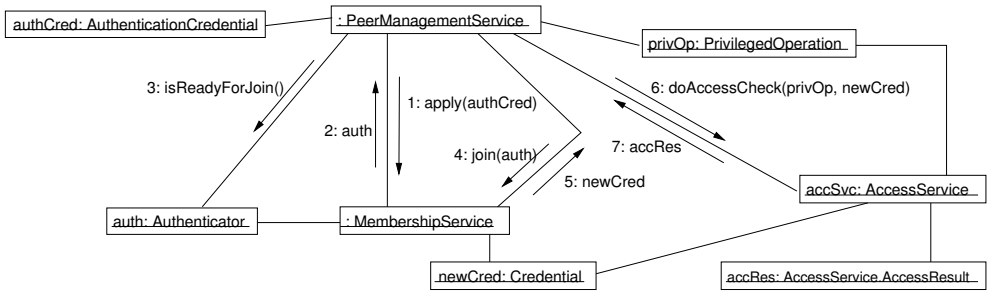

Figure 3. Collaboration diagram of peergroup authentication and authorization.

Group Charter (which is part of the peergroup advertisement) with the credentials of the peer, *i.e.* the informations provided by its CA-signed certificate. Obviously, if the certificate is not signed by a recognized CA, the peer is immediately rejected.

A more critical task is the implementation of mechanisms preventing attacks based on message flooding (see Section 3.2) and, in general, avoiding that peers perform unauthorized tasks. An AccessService implementation provided by JXTA includes the permission table in the peergroup's advertisement, making no effort to ensure that the permission table has not been altered, because an hackered peer could be able to modify the retrieved peergroup advertisement. Our solution to this problem, based on GMC control, is under development.

5 Related Work

Most prior work in group security has focused on key management and authentication [8], and on group admission control [21]. As we already stated, these building blocks are necessary for both the two main group security approaches: reputation scheme, and trust negotiation scheme.

The reputation approach, in which a peer makes decisions based on its own experience and on other peer's recommendations, was introduced and motivated in [6, 7]. The authors declare that security technology such as cryptographic algorithms for privacy, digital signatures, authentication protocols and access control methods cannot manage the more general concept of "trustworthiness". The purpose of their work is to define a distributed trust model, based on specific statements which define trust at an higher level, avoiding the ambiguity of existing descriptions. The authors introduce a recommendation protocol for trust information exchange. Other recently proposed reputation systems are [19, 13]. Even though these solutions require some form of persistent node identification (such as verifiable real-world identity), they are not suited, *e.g.*, to formal trust relationships based on legally binding contracts.

The trust negotiation approach is based on the principle that trust must be placed in a "trusted authority" [26]. In [22] it is considered the case where an ad hoc network is under the responsibility of a mother certification authority (mCA). Since the nodes can frequently be collectively isolated from the mCA, but still need the access to a certification authority, the mCA pre-assigns a special role to several nodes (called servers) that constitute a distributed certification authority (dCA) during the isolated period. The authors propose a solution to manage the dCA, called DICTATE, ensuring that the dCA always processes a certificate update (or query) request in a finite amount of time and that an adversary cannot forge a certificate. More in general, distributed GAuths are introduced in [23, 8], motivated by the volatility of group membership, which necessitates the distribution of highly sensitive operations throughout the peer group itself to ensure their availability. In particular, (t, n) *threshold cryptography* allows n parties to perform a cryptographic operation (*e.g.*, decrypt or sign a message), in a way that any t parties can perform this operation jointly, whereas, no coalition of $(t-1)$ or fewer parties can perform the same operation.

6 Conclusions and future work

In this paper we have described the current status of a the SP^{2}A project, a service-oriented architecture which take advantage of the strengths of two technologies at different levels, by relying on a P2P network as a decentralized carrier of light/simple service advertisements and on Grid service query and invocation mechanisms.

Recent improvements to the SP^{2}A prototype, namely the introduction of interest-based secure peergroups and Certification Authority peers, have been illustrated. We firstly have explained the motivations underneath the choice of introducing subgroups in the P2P overlay networks, referring to possible scenarios. Then we have illustrated the group security policies of SP^{2}A, and the mechanisms for peer authentication, peergroup admission control, authorization and transport security. Finally, we have provided some de-

tails about the prototypal implementation we are testing in our University Campus.

Several open issues remain. In our opinion, there should be different levels of CA peers, forming chains of trust. Moreover, it is necessary to introduce a Registration Authority, providing CAs the necessary information for the creation of enriched certificates (with the subject's personal attributes, allowed by the extensions introduced with version 3 of the X509 standard).

CA-signed certificates should be used not only for peer-group admission, but also for service access. Fortunately, the Globus Toolkit (which is the most used OGSA-based Grid middleware) has adopted X509.V3 certificates as the main tool for secure service interaction.

Moreover, there is the problem of malicious peer. The solutions for their exclusion range from peer degradation to GMC revocation. The GAuth (*i.e.* SP^2A admin peers) could maintain a replicated *Certificate Revocation List (CRL)*. To be effective, the CRL must be readily available to any who need it whenever it is needed and must be updated frequently. Peers which are service providers should could check the CRL each time they receive a new certificate. To avoid the delays (and possible costs) associated with directory searches, it is likely that the service provider would maintain a local cache of valid certificates and revoked certificates.

7 Acknowledgments

This work has been supported by the "WEB-MINDS" FIRB project of the National Research Ministry.

References

[1] The Glubus Alliance homepage. http://www.globus.org.

[2] Project JXTA homepage. http://www.jxta.org.

[3] Public-key Infrastructure (X.509) Working Group homepage. http://www.ietf.org/html.charters/pkix-charter.html.

[4] Single Sign-On homepage. http://www.opengroup.org/security/sso/.

[5] S/MIME Mail Security homepage. http://www.ietf.org/html .charters/smime-charter.html.

[6] A. Abdul Rahman and S. Hailes. A Distributed Trust Model. In *ACM Workshop on New Security Paradigms*, September 1997.

[7] A. Abdul Rahman and S. Hailes. Supporting Trust in Virtual Communities. In *IEEE 33rd Annual Hawaii International Conference on System Sciences HICSS-33*, January 2000.

[8] Y. Amir, Y. Kim, C. Nita Rotaru, J. Stanton, J. Schultz, and G. Tsudik. Secure Group Communication Using Robust Contributory Key Agreement. *IEEE Transaction on Parallel and Distributed Systems*, 15(5):468–480, May 2004.

[9] M. Amoretti, M. Reggiani, F. Zanichelli, and G. Conte. SP^2A: Enabling service-oriented grids using a Peer-to-Peer Approach. To appear in Proc. of the 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2005), June 2005.

[10] S. Castano, A. Ferrara, S. Montanelli, E. Pagani, and G. P. Rossi. Ontology-Addressable Contents in P2P Networks. In *WWW'03 1st SemPGRID Workshop*, May 2003.

[11] R. Chen and W. Yeager. Poblano - A Distributed Trust Model for Peer-to-Peer Networks. Technical report, Sun Microsystems, 2001.

[12] B. Clifford Neuman and Theodore T'so. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications*, 32(9):33–38, September 1994.

[13] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *9th ACM Conference on Computer and Communications Security*, November 2002.

[14] G. Drew. *Using SET for Secure Electronic Commerce*. Prentice Hall, 1999.

[15] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG, Global Grid Forum*, June 2002.

[16] S. Frankel. *Demystifying the IPSec Puzzle*. Artech House, 2001.

[17] M. Gnasa, S. Alda, J. Grigull, and A. Cremers. Towards Virtual Knowledge Communities in Peer-to-Peer Networks. In *SIGIR 2003 Workshop on Distributed Information Retrieval*, August 2003.

[18] C. Grimmer, P. Konig, and C. Schlieder. Concept Maps as a Tool for Clustering in P2P Networks. http://konzept.auriga.wearlab.de/docs/.

[19] R. Gupta and A. Somani. Reputation Management Framework and Its Use as Currency in Large-Scale Peer-to-Peer Networks. In *Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, August 2004.

[20] A. Juels and J. Brainard. Client Puzzles: A Cryptographic Defense Against Connection Depletion Attacks. In *NDSS '99 Networks and Distributed Security Systems*, February 1999.

[21] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *IEEE Symposium on Network Computing and Applications (NCA-03)*, April 2003.

[22] J. Luo, J.-P. Hubaux, and P. Eugster. DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks. Technical report, EPFL, 2004.

[23] M. Narasimha, G. Tsudik, and J. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: the Case of Membership Control. In *IEEE International Conference on Network Protocols (ICNP'03)*, November 2003.

[24] E. Rescorla. *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley, 2001.

[25] W. Stallings. *Cryptography and Network Security - Principles and Practices*. Prentice Hall, 2003.

[26] S. Ye, F. Makedon, and J. Ford. Collaborative Automated Trust Negotiation in Peer-to-Peer Systems. In *Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, August 2004.