Providing a collaborative mechanism for peer group access control

Joan Arnedo-Moreno and Jordi Herrera-Joancomartí
Estudis d'Informàtica, Multimèdia i Telecomunicacions
Universitat Oberta de Catalunya
Rb. Poble nou 156, 08018 Barcelona
{jarnedo,jordiherrera}@uoc.edu

Abstract

Peer-to-peer applications enable users to create a communications framework from scratch without the need of a central service provider. This is achievable via the aggregation of resources each one of them provide, creating a completely distributed collaborative environment. Under some circumstances, groups of users operating in a global peer-to-peer network may need to create a closed communities, limiting access to the shared resources only to its members. This can be useful for security reasons or in order to provide scoping within the global overlay network. In order to achieve this scenario, security mechanisms must be implemented. In this paper, a method for peer group access control is presented managed only by the group members in an completely autonomous way without the need of any third parties.
Keywords: Access control, peer-to-peer, distributed systems security, web of trust.

1 Introduction

Peer-to-peer is a communications model in which all involved parties must collaborate in order to provide basic services, such as content or messaging, and assumes that all peers have equivalent capabilities. This is in contrast with client/server architectures, where the centralized server manages all basic services and clients are intrinsically less capable. A major challenge when dealing with this kind of architecture is that the topology of a peer-to-peer network is dynamically changing as peers may go online and off-line. It is no longer possible to rely on a central server which is always online. That is the very reason why peers must collaborate in order for the network to correctly operate.

Usually, peer-to-peer applications are conceptualized as a global overlay network without any kind of logical segmentation or segregation as far as resource availability is concerned. Any peer may access any resource available in the network just because of the fact that it is able to reach the peer itself that provides such resource. The users may use any shared resource if they can locate it.

JXTA [1] introduced the concept of *peer group* as a collection of peers that have a common set of interests. Sometimes, the possibility of creating different (but not necessarily disjoint) groups of peers operating under the same overlay network is deemed interesting or necessary. There may be several motivations, the most typical ones being:

- *A secure environment.* Peer group boundaries permit member to access and publish protected contents. Peer groups form logical regions whose boundaries limit access to the peer group resources, in a way similar to a VPN [2], but operating at application layer.

- *A scoping environment.* Peer groups define the search scope for resource lookup. Peer groups may be used in order to limit the amount of messaging exchanges in which a peer takes part. By doing this, only those messages deemed interesting by each peer will be processed, or, at least, traffic may be limited to a manageable amount.

- *A monitoring environment.* Peer groups permit peers to monitor a set of peers for any special purpose, including heartbeat, traffic introspection, and accountability.

In order for peer groups to be able to operate effectively in a global peer-to-peer network, additional security services must be provided. This mechanisms should allow peers to be able to prove group membership to other members of the group, so they can be granted access to group resources. The interesting challenge is the fact that, because of the nature of peer-to-peer, these services must be provided by

the peers themselves in a decentralized manner and cannot rely on external parties.

It must be noted that even though the concept of peer group was defined under peer-to-peer distributed environments, it may as well be applied to the general field of ad hoc networking, since the basic principle is the same: a group of users are able to create a communication infrastructure from scratch without the need of a central service provider. In mobile wireless environments, this is equivalent to a group of nodes creating a network which cannot be accessed by other nodes even though they are within transmission range.

1.1 Paper contributions

In this paper a secure and scalable method in order to provide group access control and check group membership in a peer-to-peer environment is presented. The approach takes into account the nature of peer-to-peer networks, being fully decentralized and paying special attention to the autonomy of its members and to self-organization. Each peer should initially only manage information directly related to itself in a manner that all necessary services are provided by its members, avoiding dependency from external group entities.

1.2 Paper organization

This paper is organized as follows. Section 2 presents current proposals for access control that can be applied to peer groups. In section 3 the proposal for providing proof of group membership to peers is described. Section 4 concludes the paper and gives some ideas for further research.

2 Related Work

Group access control in peer-to-peer environments has been discussed in the scientific literature, although not always has been referred with the same name. In fact, most research has been done in the field of ad hoc networking, focusing in physical devices.

Different proposals [3, 4] use a symmetric key model. Roughly speaking, symmetric key models are the ones that use some secret information shared among the group peers in order to allow to perform group access control. This is the most obvious solution for peer group access control and proof of membership may be achieved via the direct usage of this token, simply showing its possession, or with a more suitable approach, such as using a challenge-response protocol to prove that a peer is in possession of the correct key.

However, the main drawbacks of using the symmetric model are related to key management and distribution. The shared key must be transmitted to new members of the group via an out-of-band secure channel. Furthermore, if the key has to be changed (in case of key compromise or in order to remove a peer from the group), a new key must be created and transmitted to each member of the group, which would be equivalent to creating the whole group again from scratch. On the other hand, with the symmetric model approach, any peer that is a member of the group may perform access control. However, that means that the capability of group access control cannot be limited to a very specific set of peers.

Public key models are an alternative to symmetric key models. Each peer is provided with some information which is not shared with the other peers of the group (a private key), but that will be enough to prove group membership. Peers do need to exchange some public data (a public key) that is linked to its private key, but its distribution does not need a secure channel, since it is considered computationally unfeasible to deduce the private key from the public key.

The main concern when asymmetric keys are used is that authenticity must somehow be guaranteed in this exchange since, otherwise, an active attacker may impersonate someone else's public key with his own. In order to solve the authenticity of the public keys two different approaches can be found in the literature, based on its reliance or not in a certification authority (CA).

In [5] a centralized CA is proposed. This is the most straightforward model and assigns the CA role to a single peer. However, a centralized CA approach is not a good solution in peer-to-peer networks since it would have a steep impact on availability, as the CA must be online in order for a peer to register to a group or retrieve other peer's certificates. It also provides a single point of failure in case of attack. It also goes against the basic principle of peer equality, as only the peer with de CA role may register new members. Availability may be improved by replicating the CA on n different peers, so it may withstand $n-1$ failures. However, this approach highly reduces robustness in terms of security, since the system now provides n points of failure.

In [6] a distributed CA model is used. In this proposal, the CA private key is distributed between a

specific subset of peers, but without complete replication of the CA private key. Each peer only knows a part of the secret and collaboration is needed in order to retrieve it. The CA private key is distributed among peers using secret sharing schemes [7]. In a (t, n)-threshold scheme the secret is distributed to n different peers allowing to compute the secret with the data of any t peers but obtaining no information about the secret in case $t-1$ collude. The peers among which the CA is distributed are called *server nodes*. t peers must be present at group initialization so they can jointly issue certificates to new members. Furthermore, any peer may retrieve from them authentic copies of the public keys of any other peer in the group. An additional peer, named *combiner*, is the one which adds up all operations from *server nodes* into a final result. The proposal tries to minimize impact on availability using threshold schemes, but the existence of special peers goes against peer equality in this approach. Furthermore, the work load on these peers, using the proposed protocol, is really high.

In a later improvement of the aforementioned proposal [8], the workload is lessened by letting each peer manage its own copy of the signed public key, instead of retrieving it from the server nodes each time. This proposal tries to keep peer equality by eliminating server nodes and distributing the CA to all the members of the group. Any t peers must collaborate to issue, renew or revoke public keys. The values of parameters n and t must be allowed to be changed while the system is running, since otherwise this model is susceptible to the *mobile adversary* threat. This is also important in order to allow the group to grow.

The Cornell On-line Certification Authority - COCA [9] is also based on a distributed CA model. This is the first system to integrate a Byzantine quorum system in order to achieve availability. In this proposal, defense against mobile adversaries is achieved using proactive recovery. In addition to tackling problems associated with combining fault-tolerance and security, new proactive recovery protocols were developed in this proposal. However, the proposal reduces the availability, since the protocol is more complex and efficient group communication must be guaranteed.

In MOCA [10], a different framework based on threshold cryptography is proposed. In this case, the main motivation is providing an efficient certification protocol, the MOCA Certification Protocol. This protocol reduces the amount of overhead from flooding while maintaining an acceptable level of service, introducing the concept of β-unicast, where the client can use multiple unicast connections to replace flooding if the client has sufficient routes to CA peers in its routing cache. β represents the sufficient number of cached routes to use unicast instead of flooding.

DICTATE [11] divides the CA itself in two different entities: an offline identification authority (IA) and an online revocation authority (RA). The IA authenticates the initial binding between a public key and its subject entity, and the RA keeps track of the status of certificates issued by the IA. By using this separation, compromising the online authority (which is usually more vulnerable than an offline authority) does not enable the adversary to issue certificates to new users. This proposal also uses threshold cryptography in order to delegate CA responsibility to different peers.

The proposal presented in [12] does not rely in a CA, neither centralized nor distributed, in order to manage the group. Every peer is responsible for generating and managing his own public/private key pair and other mechanisms are used in order to guarantee key authenticity. In this proposal, the concept of peer group is referred as a *troupe*. The proposal goes beyond group membership and tries to provide a distributed trust relationship between members within different groups. Members within the same group (or *troupe*) collaboratively calculate an RSA accumulator [13]. The accumulator will be used as group identifier and is considered the public key. The exponent used by each peer to create the accumulator will be the private key. Group membership may be proved via a zero-knowledge protocol for modular exponentiation [14]. In this proposal, different operations for group joining and dismissal are provided. However, it needs heavy computation and the fact that a new key is generated every time a new user joins the group means that each peer must be effectively online in order to let a new peer join the group. This model also needs a special unique peer during group member join or exclusion operations which acts a coordinator, the *troup controller*. It must be noted, however, that a troup controller is not the same as a CA, since it does not provide authenticity for public keys. It is simply a coordinator for collaborative operations and any peer may become a group controller.

Another approach is the Certificateless public key model. This model uses a public/private key scheme but no certificates or identifiers are needed. Peers will directly exchange public keys, which will act as their identifiers. The main advantage is that it obviates the need for a naming infrastructure (such as a PKI), which makes things much simpler. However, such exchange must be done over an authentic channel, since otherwise it is impossible to be sure that the received public key is really the expected one (and not from an

intruder). A protocol for public key exchange may be found in [15], which ensures authenticity via *location-limited channels*, that where introduced in the resurrecting duckling model of interaction [16], but now using asymmetric cryptography. This approach provides a mechanism in order to guarantee individual peer identity, but not group access control. Additional measures must be taken into account in order to register whether a specific peer belongs to a group or not.

Hubaux *et alter* [17, 18] present a protocol where peers have a public/private key pair and may generate and distribute their own certificates with no need of a trusted third party. Peers sign certificates for those other peers they trust, in a way similar to the PGP web of trust [19], trying to take advantage of the small world phenomenon [20] in order to encompass large groups of peers. Every peer has a list of all the certificates he has signed (peers he trusts), and all from those of peers who have created certificates for him (peers who trust him). When two peers want to authenticate each other, they exchange both lists and try to find a trusted path by merging them. In order to do this, peers must calculate several certificate validations. This model truly keeps peer equality and stays within the spirit of an infrastructureless peer-to-peer network. However, it needs out-of-band knowledge in order to prove the authenticity of public keys to be signed by each party.

Another proposal that relies on web of trust may be found in [21], which is based on node clustering. Nodes are clustered according to transmission range. However, the proposal fully focuses on identity management and is not directly applied to peer group access control, since it assumes a single global network. To sum up, CA-dependant approaches, and in particular the distributed CA models, offer a trade of between availability and security since the CA private key can be distributed between peers. Furthermore, peer equality may be fully achieved distributing such key between all peers in the group. However, the main drawbacks of this proposals are when they are applied to a dynamic group since varying the parameters in which the CA is distributed may imply recomputing all shares.

3 A collaborative approach

As discussed in the previous section, the basic concept of web of trust in PGP is ideal for a peer-to-peer environment, where all peers are autonomous and share efforts in order for the network to continue to operate, without any kind of centralized infrastructure. However, since it is a fully distributed environment where it is not possible to rely on single specific peers, central certificate repositories cannot be directly used, as PGP proposes. If peers have to be really autonomous, each one should manage only its own trust relationships and will need to collaborate and aggregate resources in order to infer trust relationships.

3.1 General definitions

Group access can be split in two steps: registration and authentication, that are defined as follows.

Registration: The process by which a new peer applies to be accepted into the group. During this process the new peer may receive any credentials (keys, passwords, tokens) that will be needed at later stages to prove group membership. It is assumed that registration is initially performed only once, and, if the process succeeds, the new peer will be considered a member of the group afterwards. A registration process may require a previous invitation or, at least, knowledge of the existence of the group itself.

Authentication: The process by which a user connects to a group, proving he is one of its members. The previous registration process provides the needed evidences for the authentication procedure. In this environment connection is equivalent to the possibility to share some resources.

The following notation will be used for the rest of paper in order to clarify the description of group membership procedures:

- Let $G = \{A_1, A_2, \cdots, A_n\}$ be the peers of the group G.
- Let B be the new peer who wants to join the group.
- Let $\Gamma^R = \{A_{i_1}, A_{i_2}, \cdots, A_{i_r}; i_j \in \{1, \cdots, n\}; \forall j = 1, \cdots, r\}$ be the registration structure within a group. That is, the set of r peers who are allowed to register a new peer.
- Let A be a peer who is already part the group and belongs to Γ^R.
- Let C be a peer who is already part the group but does not belong to Γ^R.

3.2 Group membership trust model

Authentication in a web of trust depends on a path of trusted intermediaries that will travel from the authenticating peer to the one to be authenticated. This intermediaries are defined as *introducers*. When a peer trusts an introducer, it implies a certain confidence on the capability of the introducer to provide faithful certificates. If a path really exists between both peers, the identity may be confirmed via the evaluation of the trustworthiness of each introducer's key. Otherwise the user's identity cannot really be confirmed and no real information is obtained from the web of trust.

The different trust relationships in a web of trust are usually represented as a directed graph, where edges represent trust relationships, as shown in figure 1. Here, for example, Z trusts Y, which also serves as an introducer for both X and U. That means Z may authenticate X and U via Y.

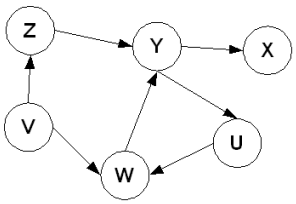

Figure 1: Web of trust representation

In order to use web of trust concepts in our model for group access control, trust relationships are translated to proof of group membership. Whenever peer A creates a trust relationship with peer B, it is acting as an introducer to B for the rest of the group members. This means that A is vouching for B's group membership to other group members. In the case presented in figure 1, Z acts as an introducer for Y, which also acts as a introducer for X and U.

In our model, two different sets of peers exist within a peer group: one composed by peers that belong to Γ^R (those which are allowed to register new members), and another composed by the rest of group members, $G \setminus \Gamma^R$. At this moment, no initial assumption is made regarding the cardinality of both sets or how a peer becomes part of each set, but it is obvious that it is necessary that $\Gamma^R \neq \emptyset$ in order for new peers to be able to join the group.

Since two different sets of peers exist within G, two different types of trust relationships are distinguished between peers, depending on which set both peers belong. Both types of trust relationships are created in the same way they are created in a standard web of trust, by signing the trusted peer public key, generating a certificate with a specific date of expiration. The created certificate specifies which kind of trust relationship it is for. Both trust relationships reinforce the fact that a peer is member of the group.

Patron relationships are established from peers which belong to Γ^R to a peer which does not belong to Γ^R. These trust relationships are unidirectional, just like the typical web of trust relationship. When peer A signs peer B's key, it will be considered to be its *patron* within the group.

Backbone relationships are established between peers which both belong to Γ^R, but are considered to be bidirectional. In order to create, backbone relationships, both peers always sign each other's public key.

An example of these different sets of peers and types of trust relationships is shown in figure 2. Grey peers are those which belong to Γ^R, and white ones do not belong. Backbone relationships between peers are marked as bidirectional arrows, whereas patron relationships are marked as unidirectional arrows.

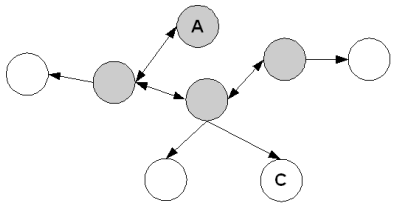

Figure 2: Initial group membership trust graph

It must be taken into account that in a global overlay network different peer groups must transparently operate. Under this model, each peer group manages membership with its own web of trust, which means that some number of independent webs of trust will coexist. In order to achieve this, each signature is binded to the specific group the patron peer is vouching, and will only be valid for operations regarding that specific group. This is done via including the group *id* into the signature. A peer may hold different signatures from the same patron, but each one binded to different peer groups.

Since peers are autonomous, each one exclusively manages only the information concerned about itself. This means that each peer will have information related to:

- Its own private/public key pair.
- The signatures from other peers.
- The list of peers trusted by him.

In order to access any other information, other peers must be asked to provide it.

Even though in PGP there are several degrees of trust (none, marginal and complete), and proposals for ad hoc networking exist which take into account trust evaluation in path validation and focus on how to assign specific values to trust [21, 22, 23], only two different degrees are initially considered under this model: whether the relationship exists or not (0 or 1 respectively).

3.3 Group membership services

Once the basic trust model has been described, how to provide the basic group membership services will be detailed. A list of group management services and dynamics may be found in [24]. However, it does not identify the difference between group registration and authentication. We will also focus on those services related to single operations, since bulk operations can be reduced to an aggregation of simple operations.

The proposal is meant to be able to adapt to a wide range of group policies and membership registration and authentication scenarios. A classification for such scenarios is presented in [25], according to the degree of involvement of peers in the registration and authentication process.

3.3.1 Group setup

In the case that a single peer decides to create a new group, no trust relationships are needed, and the peer decides whether it will be part of Γ^R or not. It must be noted, however, that in the latter case the group will never be able to grow, so it is an impractical decision.

If instead of a single peer, a set of n peers want to create a group, the process can be divided as the initial group setup described here, where the initial peer does belong to Γ^R, and $n-1$ membership registrations, explained as follows.

3.3.2 Group registration

In order for a new peer, B, to register to the group, it must apply for membership to any peer which belongs to Γ^R, such as A. If agreement is reached, a patron relationship is established and A becomes B's patron for this group. The model does not impose any restriction on deciding why a new peer is accepted into the group. This decision will be up to the group policies or the individual decision of A, and goes beyond the scope of the model.

Since signatures expire, B should renew its relationship with some patron (usually, the same as in the initial registration) when the expiration date nears.

The model also accommodates to the possibility that n peers may become patrons of B. Under this assumption, such relationships may be created once B is already part of the group or the precise moment of group registration. Any peer in $G \setminus \Gamma^R$ may ask for more patrons at any time. This assumption enables the group to provide redundancy in order to solve possible availability problems as it is shown in section 3.3.4. Furthermore, this enables implementing stricter group policies or scenarios where a minumum number of patrons must be collected before B is considered to be part of the group. A list of such scenarios may be found in [25]. Under this policy, group membership cannot be achieved with a single patron, reinforcing security at the cost of a more agile registration procedure.

3.3.3 Change of role

At some time, any peer may decide to become part of Γ^R. On this regard, the model does not enforce any rules for creation of backbone relationships, imposing no restrictions on group behavior at this level. Since it encompasses a completely open network where peers self-organize and collaborate in order to achieve basic operations, it considers that any peer which is part of the group may apply to become part of Γ^R at any time, just like it may apply to join the group. Group policies or strategies will decide if this peer is granted such role change. In fact, it may be completely possible that $G = \Gamma^R$, fulfilling total peer equality. As peers which belong to Γ^R perform more operations, it is expected that only those nodes with sufficient computer power or bandwidth will apply.

It may also be possible that a peer A decides to leave Γ^R for some reason. First of all, A must announce those peers under its patronage its intention so they can find new patrons (in the case that each peer that does not belong to Γ^R only has one patron). Until that happens, A cannot leave Γ^R.

Then, A must also tell each of those peers with a shared backbone relationship its intention. All of them are known to him, since each peer keeps the list of peers which trust him in the form of signed certificates. All of them now become its patrons, changing their relationships. If such any peer, D, is currently off-line, A will wait until that peer reconnects (while D is offline, the fact that there is a backbone relationship pending to be revoked does not affect the system, since D will never interact with the group),

or its certificate signed by D expires. Since in backbone relationships both certificates are created at the same moment, it will also mean that the certificate signed by A that D keeps will also be expired, so the relationship is completely over. At that moment, A no longer need try to contact D.

In the case that D is the only peer A has a backbone relationship with, A cannot leave Γ^R unless A wants to lose group membership.

All this process is summarized in figure 3.

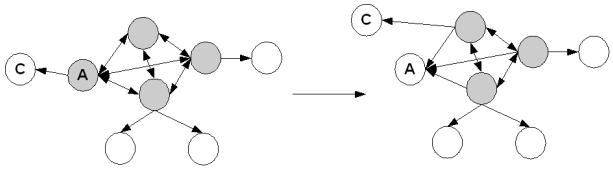

Figure 3: Change of role from Γ^R

3.3.4 Membership authentication

In order for a peer C to provide proof of membership, a trust path must be found between the authenticating peer and C. Both kinds of trust relationships are eligible when searching this path. However, this trust path must accomplish the addditional condition that all contained signatures must be binded to the peer group C is trying to access. In figure 4 a case is shown where A may be able to correctly authenticate C as a group member.

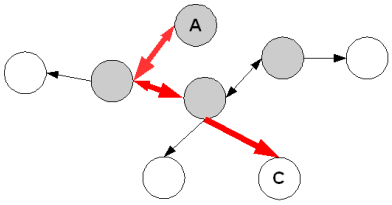

Figure 4: Finding a trust path between two peers

This model contemplates the possibility that some peers, those in $G \backslash \Gamma^R$, have insufficient information in order to authenticate, even in the case that all peers are online. The reason which motivates this is providing the model with the capability to be adapted to specific group policies that need such restriction.

In order to minimize the length of certification chains in the proposed model, when A successfully authenticates C, A checks whether it shares a backbone relationship with C's patron. If it does not exist, a new one is automatically created. As the group life progresses, peers which belong to Γ^R will eventually create a connected graph. From that moment, any peer in Γ^R may authenticate any other peer in G in two hops.

This is shown if figure 5. In *(a)* an initial state is presented. When A authenticates C, a new backbone trust relationships is created *(b)*. Eventually, full connectivity is achieved in *(c)*.

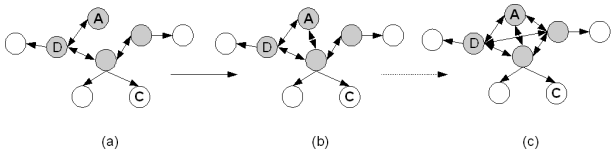

Figure 5: Complete graph backbone progress

The creation of multiple patron relationships is important in order to avoid those cases where authentication might fail because insufficient peers are online, a key issue in peer-to-peer and adhoc newtorks. For example, in figure 5 case *(a)*, should peer D somehow become unavailable, it would be impossible for C to provide proof of group membership to A, since a trust path could not be retrieved. This problem is solved in cases *(b)* and *(c)*, via the eventual growth of the backbone relationship connected graph.

4 Conclusions and further work

In this paper a method for access control in peer groups using a web of trust has been presented. The main contributions are twofold. First of all, its ability to adapt to a broad range of group policies and group membership scenarios, providing necessary capabilities in order to be more restrictive or open when needed. Also, the proposal minimizes de length of certification paths in order to avoid the validation of long certification chains, which is the main problem in several current methods.

Further work includes how to provide an effective method for efficient certificate management and membership revocation mechanisms. Another goal deemed interesting for the proposed model is how to adapt it in order to include some degree of peer anonymity within a group.

References

[1] Sun Microsystems, "Project JXTA", `http://www.jxta.org`.

[2] Huston G. Ferguson, P., "What is a vpn?", Tech. Rep., Cisco Systems, 1998.

[3] IEEE, "Standard specifications for wireless local area networks.".

[4] J. Katz, R. Ostrovsky, and M. Yung, "Efficient password-authenticated key exchange using human-memorable passwords.", *Advances in Cryptology- EUROCRYPT '2001*, pp. 475–494, 2001.

[5] CCITT, "The directory authentication framework. recommendation", 1988.

[6] L. Zhou and Z.J. Haas, "Securing ad hoc networks.", *IEEE Network Journal*, vol. 13, pp. 24–30, 1999.

[7] A Shamir, "How to share a secret", *Commun. ACM,*, vol. 22, pp. 612–613, 1979.

[8] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks.", *International Conference on Network Protocols (ICNP)*, pp. 251–260, 2001.

[9] L. Zhou, F.B. Schneider, and R.V. Renesse, "Coca: A secure distributed online certification authority.", *ACM Transactions on Computer Systems*, pp. 329–368, 2002.

[10] S. Yi and R. Kravets, "Moca: Mobile certificate authority for wireless ad hoc networks.", *The 2nd Annual PKI Research Workshop (PKI 03)*, 2003.

[11] J. Luo, J.P. Hubaux, and P.Th. Eugster, "Dictate: Distributed certification authority with probabilistic freshness for ad hoc networks.", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 311–323, 2005.

[12] S. Gokhale and P. Dasgupta, "Distributed authentication for peer-to-peer networks.", *Symposium on Applications and the Internet Workshops 2003 (SAINT'03 Workshops)*, pp. 347–353, 2003.

[13] Baric Niko and Birgit Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees.", *Advances in Cryptology Eurocrypt 97*, pp. 480–494, 1997.

[14] J. Camenisch and M. Markus, "Proving in zero-knowledge that a number is the product of two safe primes", *Advances in Cryptology Eurocrypt 1999*, pp. 106–121, 1999.

[15] D. Balfanz, D.K. Smetters, P. Stewart, and H. Chi Wong, "Talking to strangers: Authentication in ad-hoc wireless networks.", *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS '02)*, 2002.

[16] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks.", *Proceedings of the 7th International Workshop on Security Protocols*, pp. 172–194, April 1999.

[17] S. Capkun, L. Buttyn, and J.P. Hubaux, "Self-organized public-key management for mobile ad hoc networks.", *IEEE Transactions on Mobile Computing 2 (2003)*, pp. 52–64, 2003.

[18] J.P. Hubaux, L. Buttyn, and S. Capkun, "The quest for security in mobile ad hoc networks.", *MobiHoc'01: Proc. of the 2nd ACM int'l symposium on Mobile ad hoc networking & computing*, pp. 146–155, 2001.

[19] S. Garfinkel, *Pgp: Pretty good privacy.*, O'Reilly and Associates Inc., 1994.

[20] D. Watts, *Small Worlds*, Princeton University Press, 1999.

[21] Michael R. Lyu Edith C. H. Ngai, "Trust- and clustering-based authentication services in mobile ad hoc networks", *24th International Conference on Distributed Computing Systems Workshops - W4: MDC (ICDCSW'04)*, pp. 769–780, 2004.

[22] A.W. Thompson R.A. Zhaoyu Liu, Joy, "A dynamic trust model for mobile ad hoc networks", *Proceedings. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pp. 80–85, May 2004.

[23] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks", *Proc. 3rd European Symposium on Research in Computer Security – ESORICS '94*, pp. 3–18, 1994.

[24] S. Magliveras X. Zou, B. Ramamurthy, *Secure Groups Communications Over Data Networks*, Springer, 2005.

[25] Joan Arnedo-Moreno and Jordi Herrera-Joancomartí, "Identifying different scenarios for group access control in distributed.", *to be published in Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2006)*, 2006.