Maintaining unlinkability in group based P2P environments

Joan Arnedo-Moreno
Estudis d'Informàtica, Multimèdia i Telecomunicacions
Universitat Oberta de Catalunya
Barcelona, Spain
jarnedo@uoc.edu

Jordi Herrera-Joancomartí
Escola Tècnica Superior d'Enginyeria
Universitat Autònoma de Barcelona
Campus de Bellaterra, Spain
jherrera@deic.uab.cat

***Abstract*—In the wake of the success of peer-to-peer networking, privacy has arisen as a big concern. Even though steps have been taken in order to attain an anonymous communications channel, all approaches consider the overlay network as a single entity and none of them take into account peer group based environments. In this paper, we describe a method in order to maintain unlinkability in group membership authentication attempts when using peer groups relying on web-of-trust. Using this method, it is not possible to ultimately pinpoint a peer's identity despite the constraints of a group membership scenario.** [1]

Keywords: peer-to-peer, P2P, security, peer group, anonymity, unlinkability, distributed systems, web-of-trust, ring signatures.

I. Introduction

The adoption of peer-to-peer (P2P) technologies has become a promising solution to share distributed resources. Usually, P2P environments are conceptualized as a global overlay network without any kind of logical segmentation as far as resource availability is concerned. However, under some circumstances, it is desirable to segment the network into sets of peers which share common interest or services, creating *peer groups*. The main reasons for such segmentation range from restricting access to some resources to creating a scoping domain.

A lot of research efforts in the field of P2P have mainly focused towards strictly functionality issues such as scalability, efficient message propagation across the network or access to distributed resources. At present time, the maturity of P2P research field has pushed through new problems such as those related with security.

One of the desired security properties for a P2P system is anonymity, allowing users to connect to the P2P network without exposing their identity, protecting their privacy and escaping censorship. Initiatives such as Freenet [1], FreeHaven [2] or Tor [3] provide mechanisms to deploy fully anonymous P2P networks. In such systems, it is not possible to easily guess the source of messages transmitted across the network. A general survey on anonymity in P2P systems may also be found in [4].

Unfortunately, the aforementioned initiatives do not take into account peer group based environments. These proposals rely on the assumption of a flat network where peers do not form groups, and then anonymity is focused on messaging, resource access and publication. However, as we already mention, network segmentation in different groups could be useful in some applications but it would be interesting to still preserve user anonymity. That means it will be necessary to identify which peers are group members, without disclosing its identity or being able to trace peer interactions. In such scenario, an additional mechanism for anonymously proving peer group membership is necessary.

The goal and main contribution of this paper is to provide some degree of anonymity to peer groups, allowing peers to prove membership to each other without disclosing their actual identity. Specifically, the proposed method is concerned with identity unlinkability: even though members of a peer group may know the identity of its current members, it is not possible to trace authentication attempts to a specific identity or tell which have been initiated by the same peer. This is achieved with the help of ring signatures [5], [6]. Our proposal is based on a web-of-trust scenario [7], since such models heavily take into account peer equality and decentralization. Such features are important regarding the anonymity problem, since on one hand they avoid that a single peer becomes too powerful and is able to compromise the rest of the group member's anonymity, and on the other hand each member manages its own data.

The proposal assumes that peers are already provided with anonymous transport, having the capability to anonymously exchange messages at a lower layer using any of the initiatives that already exist, such as [1], [2], [3]. Otherwise, the point of anonymously proving peer group membership becomes moot as the source peer identifier is sent across the network in plain text.

This paper is organized as follows. Section II, briefly describes how group membership access control is usually attained in a peer group and the challenges it poses when applied to an anonymous environment. Following, section III provides an overview of ring signature scheme. Section IV presents how unlinkability may be attained despite the

[1]This work was partially supported by the Spanish MCYT and the FEDER funds under grant TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER CSD2007-00004 "ARES", funded by the Spanish Ministry of Science and Education.

constraints of a peer group environment. Finally, section V summarizes the paper contributions and outlines further work.

II. Related work

In this section we review the related work regarding peer group membership mechanisms and the existing proposals regarding peer anonymity.

Peer group membership approaches have been discussed in the scientific literature, although not always have been referred with the same name or taken into account the characteristics of a P2P environment.

The most basic approaches use a symmetric key model similar to that of ad hoc networks [8], [9]. This is an obvious solution for peer group access control, as proof of membership is achieved via the direct usage of this token. However, its main drawbacks are key management and distribution. The shared key must be transmitted to new members of the group via an out-of-band secure channel and changing such key is equivalent to recreating the group from scratch.

Current group membership approaches mainly focus on asymmetric cryptography. Every peer generates its own public-private key pair, which is used in order to authenticate to other peers. Most of such approaches [10], [11], [12], [13], [14] rely on a Certification Authority (CA) which generates certificates to group members, binding their public key to their identity. Those certificates may be used as proof of peer group membership, providing simplicity to peer group management, as a single trusted entity takes care of everything. However, the distributed nature of P2P networks tends to avoid relying on a fully centralized CA, using instead threshold cryptography to split the CA's private key between different peers [15].

A different approach, specially suited to P2P, is to rely on a web-of-trust instead of a single entity, such as a CA. This approach also prevents a single peer from becoming the only group manager by making use of the system's self-organization and enforcing peer equality. A proposal specifically based on group membership, though not anonymously, is presented in [16], [17]. All peers act as a fully functional CA, vouching for other peers group membership by signing certificates to them. Any peer P_i vouching for some other peer P_j's membership, is considered as its *patron*. Peers test group membership by finding trust paths (or certificate paths) [18]. A trust path is a chain of multiple certificates which validates some subject's public key, starting from the validating entity and ending in the subject to be validated.

Peer group membership approaches with the specific goal to maintain anonymity exist [19], [20]. Unfortunately, these approaches are entirely based on a centralized model, rather than a P2P architecture, where a single entity controls group membership.

Another recent proposal [21] takes into account the idiosyncrasies of peer group scenarios. However, its main goal is maintaining linkability in such a way that it is possible to identify that different authentication attempts have been performed by the same peer, which is just the opposite of our goal.

The problem of anonymous proof of peer group membership is also very similar to that of generic anonymous authentication [22], [23] and anonymous identification in ad hoc environments [24], [25]. However, there's a significant difference in our base scenario. In the cited proposals, group formation is made within the context of a user population where either a single entity generates some common knowledge to be shared within the group (equivalent to a symmetric key approach), or a flat membership hierarchy exists (as it is the case in a CA approach). In contrast, our work approaches a web-of-trust based environment, where each peer is fully autonomous and trust relationships without a single root entity must be taken into account.

III. Ring Signatures

The notion of a ring signature scheme [5] is related to that of group signature [26]. In the latter, a trusted group manager predefines certain groups of users and distributes special keys to their members. Each member can use these keys to anonymously generate signatures which look indistinguishable to other group members. In contrast, the former does not need a group manager. This is a highly desirable feature in a self-organized environment such as P2P.

Ring signatures are useful when the members are autonomous and do not want to rely on other peers. They are signer-ambiguous and provide no way to revoke the anonymity of the actual signer. It is only necessary to assume that each peer group member already holds a private/public key pair of some standard signature scheme. A ring signature is generated by the actual signer declaring an arbitrary set of possible signers, which includes himself, and computing the signature by himself using only his private key and the others' public keys.

In this scheme, the set of possible signers is called a ring. The ring member who produces the actual signature is the signer and each of the other ring members is a non-signer. The signer does not need the consent or assistance of the other ring members to put them in the ring, only knowledge of their public keys is needed. Two procedures are defined:

- $\sigma = ring - sign(m, PK_1, PK_2, ..., PK_n, SK_i)$ produces a ring signature σ for the message m, given the public keys $PK_1, PK_2, ..., PK_n$ of the n ring members, together with the private key SK_i of the i-th member, the actual signer, for $1 \leq i \leq n$.
- $ring{-}verify(m, \sigma)$ accepts a message m and a signature σ and outputs either true or false. σ includes the public keys of all possible signers ($PK_1, PK_2, ..., PK_n$).

Verification satisfies the usual soundness and completeness conditions, but since ring signatures are signer-ambiguous, the verifier is unable to determine the identity of the actual signer: in a ring of size n, probability is not greater than $1/n$. This limited anonymity is unconditional [5], since even an infinitely powerful adversary cannot link signatures to the same signer.

Ring signatures are also particularly efficient, since generating or verifying a ring signature costs the same as generating or verifying a regular signature plus an extra multiplication or

two for each non-signer. This means that the scheme is still useful even when the ring cardinality is very high.

IV. Authentication unlinkability with ring signatures

The following notation will be used in this section.

- PK_i: Peer P_i's public key.
- SK_i: Peer P_i's secret (private) key.
- $Cert_i$: One of peer P_i's certificates, containing PK_i. It must be noted that in a web-of-trust, P_i may hold several certificates (obtained from different patrons). However, they all always contain PK_i.
- $E_{PK_i}(x)$: A string x encrypted using the public key of peer P_i.

Group access control using asymmetric cryptography by means of digital certificates, is normally performed through trust paths. When some peer P_n wants to prove to some other peer P_1 that he is part of the group (P_1 authenticates P_n), all certificates conforming a trust path between P_n and P_1 are transmitted. A trust path from peer P_1 to peer P_n is a list of certificates $(Cert_1, Cert_2, Cert_3, ..., Cert_n)$, where P_1 signed $Cert_2$, P_2 signed $Cert_3$, etc. up to $Cert_n$. Consequently, P_n may be considered a group member by P_1 only if such trust path exists.

However, using this approach, the public key PK_n becomes the main constraint in order to achieve unlinkability between authentication attempts. Even in the case that $Cert_n$ does not contain any information regarding P_n's identity, the public key can be regarded as a unique identifier or pseudonym and only some degree of pseudonymity is ultimately achieved.

In order to solve this problem, we introduce the concept of *trust tree, TT* between two peers, which will be used as a means for authentication instead of a trust path, in order to solve this issue.

A trust tree between peers P_i and P_j, TT_j^i, is defined as a set of certificates $\{Cert_1, Cert_2, \cdots, Cert_n\}$ with the following properties:

- $\exists i \in \{1, \cdots, n\}$ such that $Cert_i$ is P_i's certificate.
- $\exists j \in \{1, \cdots, n\}$ with $j \neq i$ such that $Cert_j$ is P_j's certificate.
- $\forall k \in \{1, \cdots, n\}$ there exists a trust path from $Cert_i$ to $Cert_k$. As a result, it is guaranteed that a trust path exists from $Cert_i$ to $Cert_j$.

The certificate of P_j in TT_j^i is unknown to anybody but its creator: P_j. It is only guaranteed that PK_j is in some certificate within TT_j^i. Also note that, in contrast with trust paths, $Cert_j$ doesn't need to be the edge one, it may be anyone within TT_j^i. As a result, given a peer group, more than one possible TT_i^j may exist between two peers.

TT_j^i can be envisioned as a certificate hierarchy which comprises several trust paths, as shows the example in figure 1.

Since TT_j^i is a set of certificates, it contains the set of public keys $PK_{TT_j^i} = PK_1, \cdots, PK_i, \cdots, PK_j, \cdots, PK_n$. As a result, it is feasible to apply a ring signature scheme where P_j may specify the TT_j^i as the set of possible signers. Choosing a ring signature scheme is worthwile in a P2P environment as shown in section III.

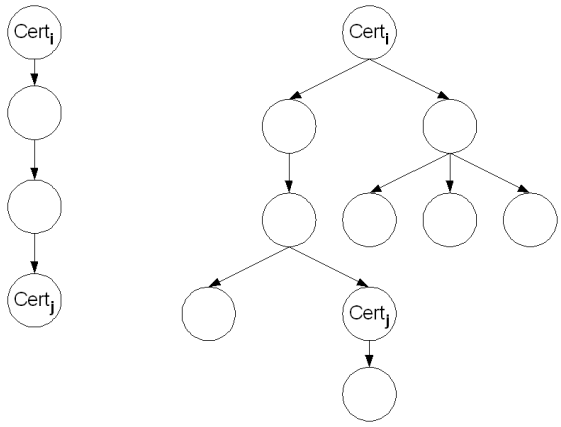

Fig. 1. Trust Path and possible Trust Tree between peer P_i and P_j

A. Authentication process

In order for P_j to anonymously proof group membership to P_i, a valid TT_j^i must be generated by P_j in advance. The authentication process then follows, as shown in figure 2.

1) A session identifier, sid, is chosen by P_j and sent encrypted to P_i.
2) P_i generates a pseudorandom nonce, r, which is sent in response to P_j. P_i stores both sid and r as an authentication transaction in progress.
3) P_j generates σ, a ring signature of r using its own private key and the set of public keys in TT_j^i.
4) P_j sends to P_i the values sid and σ, which includes both the signature algorithm result and the related the set of public keys, $PK_{TT_i^j}$.
5) P_i uses sid to identify the transaction, retrieves r and checks the signature's correctness.

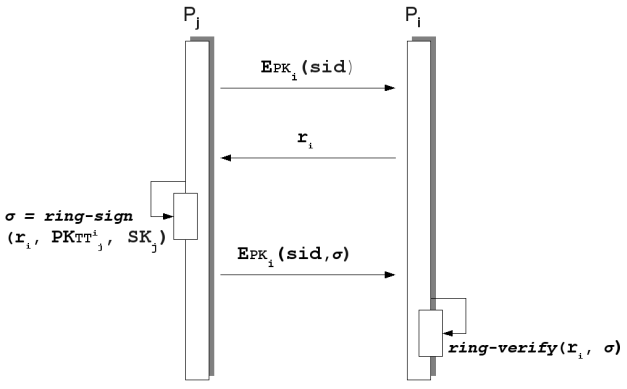

Fig. 2. Peer group membership authentication protocol

If $ring - verify$ validates, P_i checks the validity TT_j^i, as it will be shortly explained. In case that TT_j^i is valid,

P_j's peer group membership is considered proved to P_i. Short term access as a group member is granted using $E_{PK_i}(sid)$ in any further messages. During this access, interactions are linked via sid (but the identity of P_j is never disclosed). However, P_j may reset the identifier by re-authenticating. Any authentication attempt remains unlinked to the previous ones.

In order to check the validity of a trust tree TT_j^i, for any certificate from some peer $P_k \in TT_j^i$, the trust path from P_i to P_k must exist. The TT validation process is considered valid if all trust paths between $Cert_i$ and any other certificates are valid, which means that any subject within the TT is a peer group member. During this validation process, P_i stores in a local cache correct trust paths. Just certificate subjects are stored, not the whole certificate. This cache can be used in further TT validations in order to speed up the process by first looking up if a trust relationship between P_i and some other peer has already been checked in previous TT validations. And advantage of keeping this cache is the fact that it can be used in the validation process of any TT within the peer group.

In this proposal, each possible signer for a given TT (each certificate subject within the TT) becomes the anonymity set [27] for each authentication attempt. The signer's identity becomes hidden within all subjects in the TT, however, since during the validation process it has been guaranteed that all subjects within the TT are group members, it can also be guaranteed that the signer is a group member.

A TT is used instead of a simple trust path in order to both dynamically expand the cardinality of the anonymity set and avoid clearly pinpointing the actual signer (which is usually the owner of the edge certificate). Using a standard trust path as a ring signature signer set is not desirable since its cardinality, once established, will usually remain static. Since the degree of unlinkability relies on the number of possible signers, in some instances, such cardinality may become too narrow to be considered acceptable. TT's take advantage of some other peer's long trust paths in order to increase the cardinality of the signer set. Furthermore, at each authentication attempt between P_i and P_j, the later is not bound to always using exactly the same TT. P_j is able to generate different TT's which are considered valid, resulting in diverse anonymity sets which may be used at authentication attempts.

Even in the worst case scenario where P_i directly trusts P_j, an acceptable TT may still be produced. It is enough to include some other peers which conform trust paths from A, even though B does not appear in those paths, as shown in figure 3 as an example. Even in this scenario it is still possible to chose which is the cardinality of the anonymity set, up to the full peer group's cardinality.

B. Trust tree generation

A TT cannot be generated by arbitrarily collecting and setting together member public keys, which is the assumption in a basic ring signature approach, since it must be ensured that all the included public keys conform some trust path from the authenticating peer. No unconnected certificate may be included.

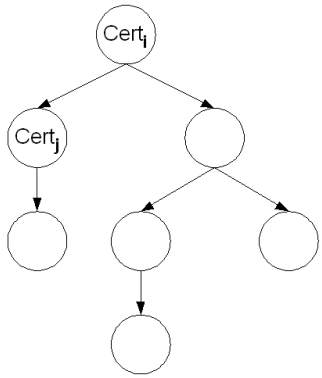

Fig. 3. Possible TT_j^i where P_i directly trusts P_j

There are several methods to obtain the necessary certificates in order to generate a TT by taking advantage of a web-of-trust based peer group membership operation. However, it is highly desirable that certificate retrieval is performed along standard network operation, since the impact of using TT"s on network performance is minimized, and most important, other peers then cannot know whether certificate retrieval is being requested in order to compose TT's or just in order to routinely operate.

Feasible methods for some peer P_j to retrieve certificates from other group members, other than just directly requesting them, are:

- Whenever P_i is requested a certificate by someone interested in becoming group member, acting as a patron following the approach in [17]. The generated certificate may be stored for later use for TT creation. An advantage of this option is that all certificates will be useful for TT generation, since all of them conform a valid trust path from P_j. Using this method is also an incentive to become a patron, since the most certificates P_j generates, the easier it is for him to generate large anonymity sets.
- In web-of-trust environments, peers already know its patron's certificates.
- In onion routing anonymous networks, which are the most popular ones, the sender establishes the message path by retrieving the certificate of each peer in the desired path. This is necessary to create each message encryption layer.That means that some method to directly retrieve other peer's certificates must exist. An advantage of this method is that both onion routing operation and TT generation make use of the same information.

The proposal in [17], in fact, defines a protocol which specifically looks up trust paths within the peer group and is able to retrieve the certificates of other group members. This capability allows to retrieve any certificate within the peer group and easily join trust paths in order to create TT's.

As shown at the beginning of the authentication process in

subsection IV-A, TT's may be generated in advance. It does not need to be created at the precise moment before initiating authentication. Peers may slowly generate a large TT as they operate within the peer group and learn about new certificates. Different subsets from a large TT, which could amount to the whole peer group in the best case scenario, may be used at each authentication attempt, as shown in figure 4. There are two main reasons for not using the whole information and just using subsets: avoiding sending large TT's across the network, which may impact performance and slow TT validation, and preserving the security of the scheme, as will be explained in section IV-C.

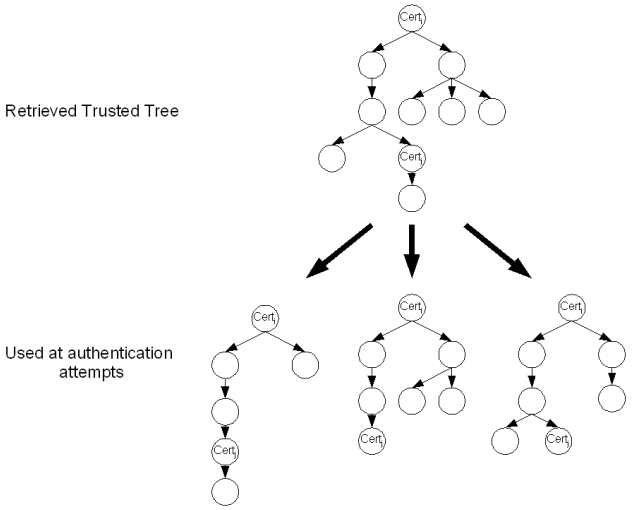

Fig. 4. Using a large TT to generate smaller TT's

C. Security analysis

In this section attacks on anonymity are analyzed. This analysis will focus only on the authentication process, since an anonymous transport method is assumed, as stated in section I. For that reason, all strengths and weaknesses of the anonymous networks will be inherited. Common attacks in anonymous networks such as the predecessor attack [28], for example, are not discussed, being completely concerned with message relay.

- *Trusted tree reuse:* A big concern in this proposal is the reuse of TT's. If the same TT is always used, it will ultimately become like an identifier and it will be trivial to link different sessions. For that reason, the same TT should not be reused in different sessions. This problem is solved with the generation of a bigger TT and then only using smaller subsets.
  However, the proposed scheme still minimizes this attack to some degree, since it cannot be guaranteed that the same trusted tree, received several times by the authenticating peer at different group membership authentication attempts, was sent from the exact same peer. Two peers may generate the same TT, and still follow the properties listed in section IV, since a given TT may be used by any peer whose public key is contained in such TT.
- *Timing attack:* In a timing attack, an authenticating peer may try to link different sessions by comparing response timings in the authentication process (message round trip time). Although anonymous networks already take this attack into account, this problem can also be solved by purposely delaying responses a random amount of time in the authentication protocol.
- *Intersection attack:* This attack is complementary to that of TT reuse. The authenticating peer may try to compare different TT's used by the same peer, intersect all of them, trying to decrease the number of suspects. An underlying anonymous network does constrain this attack only to the authenticating peer (it cannot be attempted by a middle point peer). However, this attack is countered since it is not possible for an attacker to know which TT's come from the same peer in order to compare them. All authentication attempts (as well as sessions) are completely independent and no information is sent along each one in order to relate different attempts to each other. Given a set of different TT's in which an specific public key appears in all of them, it cannot be guaranteed that some of them came from the same peer.

V. Conclusions and further work

A method to maintain unlinkability between authentication attempts in peer groups based on web-of-trust has been presented. This is achieved by using ring signatures and introducing the concept of trust trees as the means order to transport public keys as well as providing an anonymity set for the signer, which may be constructed according to its own needs and may be increased or changed over time.

The main contribution of this proposal is the capability of each individual peer to chose its trust anchors into the peer group, instead of being forced to use a specific one (such as a CA or group manager). We consider that this freedom of choice is extremely important in an environment where privacy is highly valued. Furthermore, it is faithful to a pure P2P approach, relying on peer autonomy and self-organization.

Furthermore, the proposed method also nicely meshes into peer group operation in web-of-trust based scenarios in two different ways. First of all, trust tree generation does not force the use of additional protocols, taking advantage of standard data exchanges in order to retrieve public keys. This fact improves its performance and minimizes threats from a passive attacker. Second, the link between trust tree size and the anonymity set directly rewards those peers who decide to act as patrons within the group, providing an incentive peers which act as such. In fact, in a web-of-trust based environment, it is very important that as many peers as possible agree to act as patrons.

Currently, research is at its initial stages, the anonymous authentication model being just set. Further work includes implementing the model in order to assess its behaviour and

performance under a real group based P2P middleware. A good candidate for this is JXTA [14].

REFERENCES

[1] Wiley B. Clarke I., Sandberg O. and T.W. Hong, "Freenet: A distributed anonymous information storage and retrieval system.", *Lecture Notes in Computer Science*, p. 46, 2002.

[2] Freedman M.J. Dingledine R. and D. Molnar, "The free haven project: Distributed anonymous storage service.", *Lecture Notes in Computer Science*, p. 67, 2001.

[3] Mathewson N. Dingledine R. and Syverson P., "Tor: The second generation onion router", *Proceeding of the 13th USENIX Security Symposium*, August 2004.

[4] X. Ren-Yi, "Survey on anonymity in unstructured peer-to-peer systems", *Journal of Computer Science and Technology*, vol. 23, no. 4, pp. 660–671, July 2008.

[5] Tauman Y. Rivest R.L., Shamir A., "How to leak a secret", *Lecture Notes in Computer Science*, vol. 2248, 2001.

[6] Harn L. Ren J., "Generalized ring signatures", 2001.

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

[8] IEEE, "Standard specifications for wireless local area networks.", 1999, http://standards.ieee.org/wireless.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] Sun Microsystems, "Project JXTA", 2001, http://www.jxta.org.

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

[16] Joan Arnedo-Moreno and Jordi Herrera-Joancomartí, "Providing collaborative mechanism for peer group access control", in *Proceedings of the Workshop on Trusted Collaboration*. 2006, pp. 1–6, IEEEPress.

[17] Joan Arnedo-Moreno and Jordi Herrera-Joancomartí, "Collaborative group membership and access control for jxta", in *COMSWARE'08: Proceedings of 3rd International Conference on COMmunication System softWAre and MiddlewaRE*. 2008, IEEEPress.

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

[19] Patrick Tsang, Man Ho Au, Apu Kapadia, and Sean Smith, "Blacklistable anonymous credentials: Blocking misbehaving users without ttps", in *Proceedings of CCS 2007*, 2007.

[20] Peter C. Johnson, Apu Kapadia, Patrick P. Tsang, and Sean W. Smith, "Nymble: Anonymous ip-address blocking", in *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, 2007.

[21] Patrick Tsang and Sean Smith, "Ppaa: Peer-to-peer anonymous authentication", *Applied Cryptography and Network Security*, pp. 55–74, 2008.

[22] Franklin M. Boneh D., "Anonymous authentication with subset queries", in *ACM CCS 1999*, 1999.

[23] Stadler M. Camenisch J., "Efficient group signature systems for large groups", in *Crypto 1997*, 1997.

[24] Persiano G. De Santis A., Di Crescenzo G., "Communication-efficient anonymous group identification", in *In 5th ACM Conference on Computer and Communications Security*, 1998.

[25] Nicolosi A. Dodis Y., Kiayias A. and Shoup V., "Anonymous identification in ad hoc groups", in *Advances in Cryptology - EUROCRYPT 2004*, 2004, vol. 3027, pp. 609–626.

[26] Chaum D. and Van Heyst E., "Group signatures", *Advances in Cryptology EUROCRYPT 91*, vol. volume 547 of Lecture Notes in Computer Science: 257-265, 1991.

[27] Khontopp M. Pfitzamnn A., "Anonymity, unobservability, and pseudonymity - a proposal for terminology", in *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*. 2001, pp. 2–9, Springer-Verlag.

[28] Levine B.N. Wright M.K., Adler M. and Shields C., "The predecessor attack: An analysis of a threat to anonymous communications systems", *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 4, pp. 489–522, 2004.