An Access Control Protocol for Peer-to-Peer Applications using Asymmetric Cryptography

Sebastian Voigt

University of Hannover
Faculty of Electrical Engineering and Computer Science
Institute for Systems Engineering – Computer Architecture and Operating Systems
Appelstr.4
30167 Hannover
voigt@sra.uni-hannover.de

Abstract: Virtual rooms and common information spaces are being used more and more frequently. In addition to internet-based solutions, ad-hoc InfoSpaces have been proposed. Their security has not been sufficiently investigated. This paper proposes a new protocol for peer-to-peer data exchange with support for access control for operations in the virtual rooms. Access rights for operations are kept in access control matrices and the security layer allows only operations from clients which are marked as allowed in the access control matrix. The protocol is based on the idea of hiding the identity of all peer-to-peer participants. Thus this protocol offers anonymity and full peer-to-peer support using asymmetric cryptography to distribute the access control matrices. This works without any central authority.

1 Introduction

As anonymity becomes more and more important these days, the users of peer-to-peer applications want to benefit from it as well. Because the amount of exchanged data and of connections is increasing fast, the need for hiding the identity grows too. Supervision of access control requires authentication. Therefore, if a user wants to use security or access control in an application, he has to hand over his identity information to other peer-to-peer applications. The identity information exchange prevents anonymity. Nevertheless, future peer-to-peer software has to help users to protect their privacy against attempts to log connection records.

There are many approaches to the “role-based”-, “id-based”- and “credential-based”-access control research areas. The resulting techniques are used to ensure security within peer-to-peer applications. A basic function within all these research areas is authentication, which requires some kind of identity information exchange. This means that these techniques are not a good solution to the problem of anonymity in peer-to-peer systems, because they demand to reveal the identity of the client for every connection or operation. There are these days a number of interesting peer-to-peer applications like Skype [Sk] or Jxta [Jx] that address security. In both cases the clients have identifiers which are implemented with asymmetric key pairs. The systems rely on trust chains to a central or to a rendezvous server. The communication sockets are secured with

symmetric encryption, but in neither of the cases the clients are anonymous. Both models assume that there is a central authority, that doesn't allow the peer-to-peer networks to get highly dynamic.

The paper is divided into five chapters. The next chapter discusses the problem of access control and anonymity for peer-to-peer applications. The third chapter presents a peer-to-peer application named InfoSpaces that was implemented by the author. The last chapter before the conclusion will describe the implementation of the security protocol.

2 Idea and solution

The aim of this project is to create a security layer for peer-to-peer applications that supports anonymity. This security layer must work in a truly peer-to-peer manner and it should be possible to add it to existing peer-to-peer applications. There must be no central authority and no central functionality involved.

In order to ease management and usability, we use access control matrices to describe access rights in virtual peer-to-peer rooms. Access control matrices were first introduced by Butler W. Lampson in 1971 [La74]. They characterize the rights of each subject with respect to every object in the system. In our work an access control matrix that describes all access rights to operations in a peer-to-peer room will be attached to it.

In order to assign access rights to clients, there must be a limited time interval in which the client has to show its identity information. The client that wants to create an InfoSpace peer-to-peer room must know to which clients it will give access to. During this time the clients have to show their temporary identity. To avoid "man-in-the-middle" attacks, the identity can be double-checked by users over another communication channel (e.g. by conversation). The double-checking is used to associate the identifier of a client with a real user situated physically nearby the creator of the room. When the entire access control matrix is complete, it can be distributed to other participants of the room. After distribution of the matrix, the clients can change their identity and still continue to work in this InfoSpace. We decided to use access control matrices for specification of access rights because they are well suited to this area, although they were developed for server-based systems. In addition we need a protocol for the distribution of the matrices in the peer-to-peer environment.

The idea is to apply the traditional paradigm of distributing and using physical keys to the world of software. A traditional key can be used for the door of a house or a garage etc. Each key fits exactly into one key lock, but there can be more keys that fit into one key lock. This is a solution to the demand for anonymity in peer-to-peer applications because the key holder can use the key for authorization, but he doesn't have to show his identity information anymore. More than one client can hold a copy of a key. In our case the key and the key lock are represented by a pair of asymmetric keys. The creator of an InfoSpace can set up access rights defined in the access control matrix for the peer-to-peer room. The access control matrix is then translated into pairs of asymmetric keys and the access rights are distributed using them. The creator doesn't need to provide any

server functionality in relation with other clients, because every participant can authenticate other participants by itself for each operation, using the distributed keys and key locks. This has the advantage that no central authority and no central server are needed. Thus the peer-to-peer security layer that we propose supports anonymity and server-free peer-to-peer rooms with access control. This layer can be added to existing peer-to-peer applications.

3 Application: InfoSpaces

The security scheme presented in the previous chapter is applicable to any distributed peer-to-peer communication. We demonstrate its feasibility in the context of an InfoSpaces application developed by the author. InfoSpaces are communication rooms for data exchange based on the idea of Linda tuplespaces [Ge85]. An InfoSpace is an equivalent of physical communication channels. Every communication partner is allowed to drop information into an InfoSpace. Information offered in the InfoSpace can be consumed by other participants. The application has a simple GUI which is easy to use and allows a few operations only: copy in, move in, copy out and move out. Out-operations drop objects into InfoSpaces. As a result references to objects are sent to all other participants, which can execute in-operations on them. For this, they query the holder directly to request for the transmission of the object. The transfer of an information object is only started after at least one out- and one in-operation occurred. For more information please refer to [Br03].

4 Protocol

An underlying goal of the security layer is to prevent eavesdropping. This is done by encrypting all connections used in the peer-to-peer network. With this basic protection secret information (the keys in this case) can be safely exchanged. Furthermore no other client can overhear the connection. Due to the short transmission times we have chosen a symmetric encryption to protect all connections. At the moment there are some protocols that support symmetric encryption and secure exchange of a symmetric key at the same time (e.g. SSL [DC99] and Diffie-Hellman [He02]).

Compared to other techniques like e.g. “id”-based protocols, our clients don’t possess a pair of asymmetric keys for showing and proving their identities. The protocol uses a temporary identifier for each client, which can be changed easily.

The creator of an InfoSpace uses access control matrices in order to specify the access rights when it creates that InfoSpace. These matrices are then translated into keys which are being distributed to other clients over the peer-to-peer connections. Due to the length limitations of this paper we can only describe the three most essential communication patterns (creating and distributing an announcement, authorization and publishing) used by our protocol, as follows:

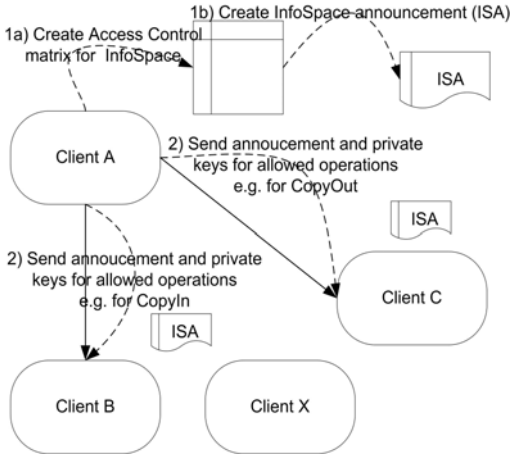

Figure 1 Distribution of an InfoSpace announcement and the corresponding keys

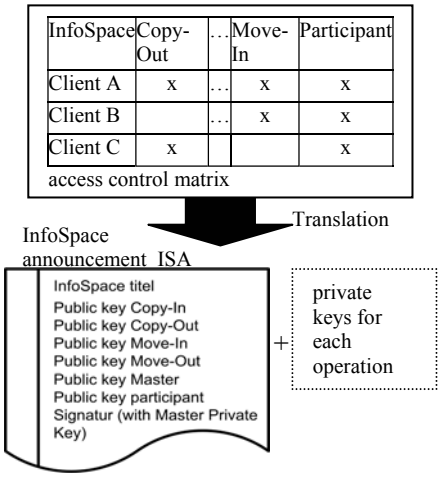

Figure 2 Translation from matrix into announcement and keys

Figure 1 illustrates the creation procedure for a new InfoSpace. At the beginning the creator of the InfoSpace (Client A) sets up an access control matrix. It uses the matrix to describe which other clients have access to certain operations. The creator then uses the identifiers mentioned before to recognize other clients (e.g. by conversation). A master of an InfoSpace is a client that can set all rights in an InfoSpace. Each creator is certainly one of the masters of the InfoSpaces it created. The matrix is translated into keys (see Figure 2). The security layer generates the necessary key pairs. It creates one key pair for every operation that is possible in the InfoSpace and two additional key pairs, one for the participation and one for master access. Subsequently the client creates an InfoSpace announcement (ISA) containing the public keys from all key pairs and the description of the InfoSpace (see Figure 2). The security layer distributes the InfoSpace announcement to every client intended to work with the new InfoSpace. Furthermore the private keys for all operations a client is allowed to perform are sent to that client and this step is taken for each such client. Every client that is intended to have rights in the new InfoSpace has received to this moment an announcement of the InfoSpace and several private keys for the operations it is allowed to perform. An intruder (Client X) that would like to get access to the newly created InfoSpace will get no information (no announcement and no private keys) about it and it will not be able to access the InfoSpace. The announcement is signed with the private key from the key pair for master access. The public key for master access is contained in the announcement, so that every client can check the signature of updated, newly distributed InfoSpace announcements in order to make sure that the announcement was distributed by the creator. This and other masters are the only ones that have the private key from the key pair for master access. After the distribution of the ISA and of the private keys, the other participants do not need to interact with the masters anymore (no single point of failure).

Since the InfoSpace announcement contains no information about which rights have been awarded to which clients, which means after all that there is no information about

who should have access to the InfoSpace, each participant is forced to control itself, whether another client is a legal participant of this InfoSpace. However this step is only necessary when it wants to send it information or data concerning this InfoSpace. Thus this authentication is not immediately necessary after establishing the connection.

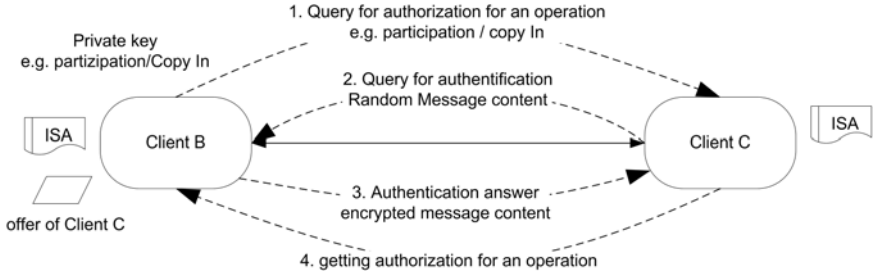

Figure 3 Authentication of operations

Figure 3 shows the steps that need to be taken by a client in order to work in an InfoSpace, while obeying the access control. The authentication for the participation to an InfoSpace is done as follows (Figure 3): A client (Client C) queries another client (Client B) to check whether it is a participant of the InfoSpace (not shown in the figure). Client B answers with a query for authorization for participation (1.). If Client B is a participant, then it has a private key and it can prove its right to participate. With this key, it encrypts the random message that client C has sent to it before (2.) and answers the query this way (3.). Client C checks whether the message was correctly encrypted with the private key, by using the public key from the announcement and knows if this participant is allowed to receive messages in this InfoSpace. Notice that the clients don't use their identifier to authenticate themselves every time. Thus no connection records can be generated.

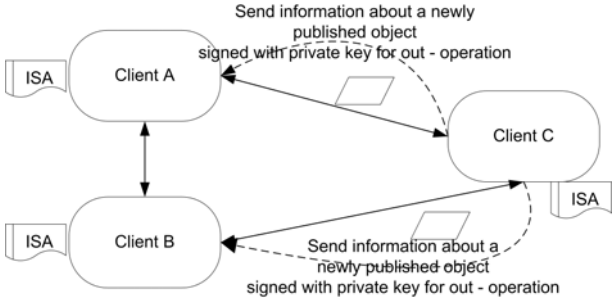

Figure 4 Publishing objects with the security layer in a peer-to-peer network

Figure 4 shows the procedure for publishing an offer for an object. An offer indicates that a client wants to share this object in the InfoSpace. Client C uses the private key corresponding to this publishing operation (copy/move OUT) to show other clients that it has been authorized for this operation (the offer is signed with the private key). The client sends the offer only to those clients, which are authenticated as participants of the InfoSpace (see above). Those clients (Client A+B) that receive an offer must check the

signature with the public key from the InfoSpace announcement. Offers without a signature or with a false signature are discarded by the security layer automatically. Thus a client that is not admitted due to a missing key cannot publish offers in this InfoSpace. Data transfer in a peer-to-peer room is controlled as follows (see Figure 3): client B who wants to accept an object offer needs an authorization for that operation. This authorization was given to it as a private key corresponding to this operation. The authorization works the same way we explained the authorization for participation. In this case the key pair for this operation (e.g. copy in) is used for the authorization. If client B can answer by encrypting with the correct private key, client C knows that client B is allowed to do this operation and it can initiate the data transmission of the object.

The implementation shows that asymmetric encryption can be used to protect peer-to-peer rooms because standard computers can process 512-bit RSA-keys with only a small time penalty (time for the creation of an ISA, including 6 key pairs and a signature: ~800ms; time to verify the signature: ~10ms on an Intel P4-M 2GHz with an Java implementation). This key length is secure because the keys are only used briefly.

5 Conclusion

In this paper we have described a new protocol to address the peer-to-peer problem of protecting virtual rooms. This protocol satisfies the demand for anonymity at the same time. The proposed security layer can be added to existing peer-to-peer applications to support access control to operations within peer-to-peer rooms like InfoSpaces. The security layer works without any central authority by using asymmetric cryptography in a new way. This allows all clients to hide their identity during every operation. The paper explains how asymmetric cryptography is used to attain these goals. Due to the short length of the paper we can not discuss the proposed approach in more detail. In further work we will propose a solution for simple group management and to support more than one master per InfoSpace. We will also show that the disappearance of the master or of the creator of an InfoSpace will not disturb the work within the InfoSpace.

References

[Br03] Brehm, J.; Brancovici, G.; Müller-Schloer, C.; Smaoui, T.; Voigt, S.; Welge, R.: An InfoSpace Paradigm for Local and ad hoc Peer-to-Peer Communication : On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, Lecture Notes in Computer Science Volume 2888 / 2003, Springer, Berlin

[DC99] Dierks, T.; Allen, C.: The TLS Protocol; http://www.ietf.org/rfc/rfc2246.txt

[Ge85] Gelernter, David: Parallel Programming in LINDA, Technical Report 359, Yale University Department of Computer Science, Jan. 1985

[He02] Hellmann, M.E.: An Overview of public key cryptography; IEEE Communications Magazine, 50th Anniversary Commemorative Issue/May 2002

[Jx] http://www.jxta.org

[La74] Lampson, B.W.: Protection. ACM SIGOPS Operating System Reviews, Vol. 8, issue 1

[Sk] http://www.skype.com