Securing Groupware for the Internet

Coulouris, George

ISSN 1369-1961

Department of
Computer Science

Technical Report No. 752

Securing Groupware for the Internet

George Coulouris

QUEEN MARY
AND WESTFIELD COLLEGE
UNIVERSITY OF LONDON

September 1998

Securing Groupware for the Internet

George Coulouris
Distributed Systems Research Group
Department of Computer Science
Queen Mary and Westfield College
London E1 4NS
Email: George.Coulouris@dcs.qmw.ac.uk

1. Introduction

Groupware is software that supports cooperative work. Cooperation depends upon the sharing of information. For the purposes of this article we take information sharing to mean that some users need to view the shared information and that some of them (and not necessarily just one user) need to alter or add to the shared information.

The Distributed Systems Research Group at Queen Mary and Westfield College is engaged in a EU-funded research project in partnership with several research institutes and companies to develop a distributed software platform for groupware applications. The project team includes system researchers, application researchers, application software developers and vendors. The 3 year PerDiS project is at its half-way stage and a preliminary version of the platform has been developed and released for evaluation [PP98]. The main goal of the project is to develop a platform that will enable existing CAD applications to be used in a cooperative manner in the building and construction industries. The platform supports information sharing by the replication of relevant data objects in the local memory of each computer where users access the PerDiS system.

Buildings are usually designed and constructed by a consortium of companies. We can therefore expect users in several different companies to cooperate on the design and construction of each building. Such a consortium is often referred to as a *virtual enterprise* . A characteristic of a virtual enterprise is that while the participating organisations cooperate on one project they are likely to be competitors on many others. Furthermore, the geographically distributed nature of such a consortium dictates the use of the Internet as the means of communication and information sharing between the participating organisations.

These characteristics are typical of real-world groupware applications. We can expect virtual enterprises to be a commonly-occurring context for cooperative work in many areas of commerce. Wherever they occur, similar security issues will arise. In this article we identify the security issues in more detail and describe our approach to the provision of a secure environment for cooperative work on the Internet.

The architecture of the PerDiS platform is illustrated in Figure 1. Many of its specific features are beyond the scope of this article. Detailed descriptions of the non-security aspects of the architecture can be found at the Web sites listed in the footnote.

It should be noted that the PerDiS platform is implemented as a 'middleware' layer that is divided into two parts occupying two separate protection domains within each machine. The PerDiS Daemon provides a distributed service for the replication of shared data in the memories of individual machines as required. It occupies a protected address space that is not accessible to application software. The User Level Library shares its address space with the application software and provides a support layer that interfaces the application to the PerDiS platform.

The protection of shared data and the management of secure communication should be a responsibility of the PerDiS middleware platform rather than the various application programs used in the cooperative activity. Hence all of the protection and secure communication mechanisms of the PerDiS platform must be located outside the address space of application programs, in the PerDiS Daemon.

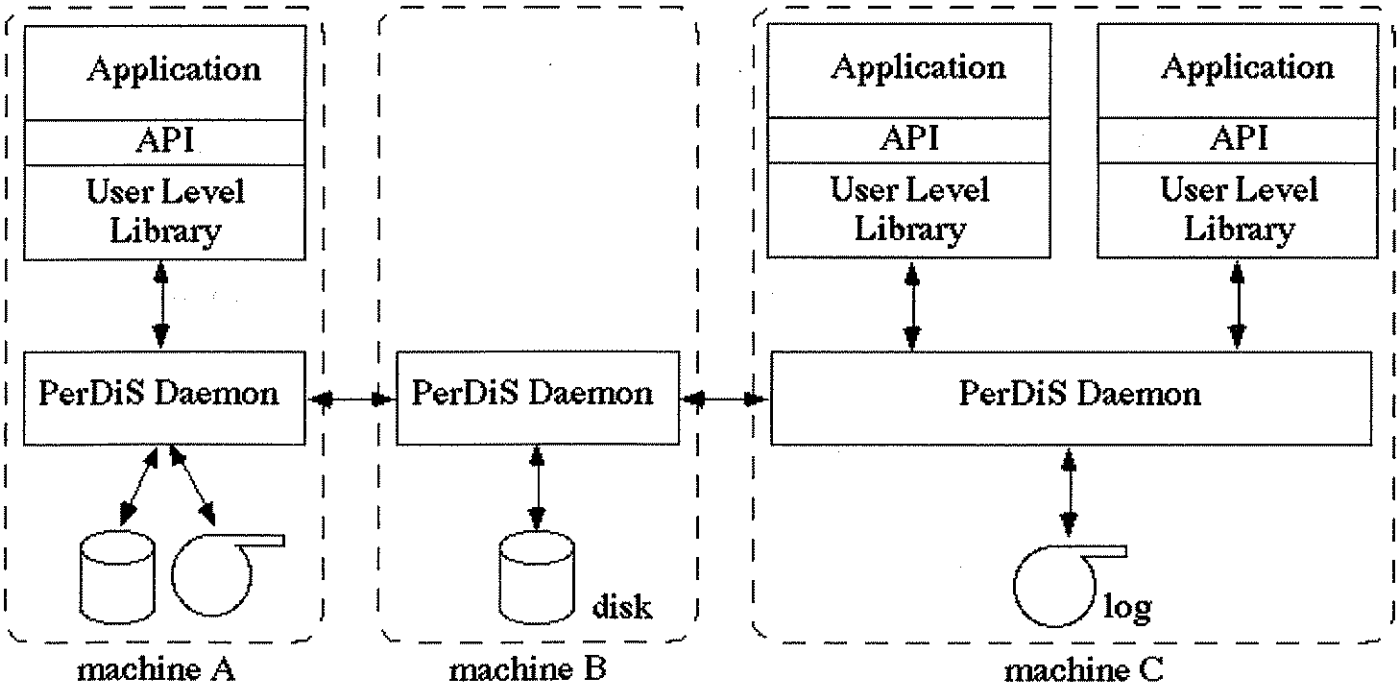

Figure 1. The Architecture of the PerDiS distributed platform.

2. Security issues

2.1. Secure communication

The PerDiS platform is a distributed system operating in an open network environment. Security threats will arise both within the individual organisations that make up a virtual enterprise and from outside. Hence the following security techniques must be applied whenever application data is security sensitive (i.e. in most cases):

Signed messages	— To ensure integrity of message contents; — to authenticate the origin of messages; — to authenticate the identity of the user responsible for a request, which is essential for the protection of objects.
Encrypted messages	— To ensures the secrecy of message contents.
Nonces or timestamps	— To ensures freshness of messages.

In practice, message signing and authentication is mandatory in PerDiS and the encryption of data is optional. The cryptographic basis for these techniques is well-established [SCHN96]. But the differences in performance between public and private-key cryptography require careful design of the distributed platform to enable appropriate tradeoffs and optimisations. This topic is discussed further below.

Trust between components of the distributed system platform

A virtual enterprise is a cooperative action between two or more companies or organisations with a defined goal. But the cooperating companies are likely to have different system administration staff and security policies. And as we have already noted, they may well have conflicting competitive interests outside the defined area of cooperation. For these reasons, the hardware and system software (i.e. operating system and PerDiS Daemon software) through which data is shared cannot be equally trusted by all because that would imply mutual confidence in the system administration procedures between all partners in a virtual enterprise.

These factors have led us to adopt a two-level trust model based on trust domains. The scope of a trust domain is configurable but it is typically defined as the local network of a company (or perhaps a division or department in the case of larger organisations). It represents a set of machines with a uniform system administration and security policy. Within each trust domain, we base authentication and secure communication on a pair of session keys (one for signing, one for encryption) that are shared between the PerDiS Daemons on the machines in the trust domain and changed relatively infrequently (e.g. daily). The trust assumption on which this is based is that a system administrator within the organisation takes responsibility for the soundness of the hardware and system software on all the machines involved.

This assumption cannot be made for the machines used in other companies. Instead, we base trust on the public keys of individual users who participate in a cooperative action (which we call a task). The designated users participating in a task in the different organisations comprising a virtual enterprise trust each other and by extension, they trust the software running on each other's machines. That trust is based on the assumption that the users participating in a task take responsibility for the soundness of the hardware and software that they use. This weaker trust assumption results in higher costs for the implementation of secure communication. We establish secure channels based on public keys in a manner analogous to that used in SSL [NETS96].

Readers familiar with the work of Lampson, Abadi *et al.* on the theory and practice of authentication [LABW92, WABL93] may wonder why we do not adopt a scheme similar to theirs, in which trust is bootstrapped through all layers of hardware and software by checking signatures on system components. The answer is that we do not have the luxury of working with signed hardware and operating system code. Nor do we expect groupware developed for open network environments such as the Internet to enjoy that luxury for the foreseeable future.

Our two-level trust model (within and between trust domains) enables the system to be tuned for security on a per-task basis, trading-off security for performance. This is achieved by varying the scope of trust domains. At one extreme, each user's machine can be defined as a separate trust domain, forcing the establishment of secure channels based on the user's public key for all communication. At the other extreme, for applications that have low security sensitivity, a single trust domain might be established across all of the partners in a virtual enterprise, enabling the use of a single shared key for all communication.

The two-level trust model is also likely to have an impact on the provision of audit trails. The user-based authentication of messages will enable simple message logs to meet the need for audit trails. But when a single shared key is used for authentication by several machines, an additional signature would have to be added to the logs to enable auditing down to the level of individual users.

Secure communication between system components

In a distributed system such as PerDiS, network communication is the means by which system components interact. At the commencement of the project, we had assumed that the provision of a secure communication layer for this purpose would present few opportunities for innovation, since there are several well-known solutions to the distribution of keys between mutually suspicious host computers and their use to establish secure communication channels. SSL [NETS96] is probably the most widely known and it provides a sound, well-understood mechanism for the establishment of secure channels between a pair of processes with no requirement for prior knowledge of each other.

In practice, to establish communication between previously unconnected processes we adopt a similar approach to SSL (and we have adopted SSLeay, a public domain implementation of SSL [HY98]). But the trust model outlined in the preceding subsection has led to a more general approach which is described in [RCD98]. It provides an efficient solution for secure communication in a wider range of situations than SSL while still confining decisions about the choice of a secure protocol to the secure communication layer.

2.2. Protection and access control

We have undertaken some case studies of user requirements for the protection of data in cooperative work [CD94]. One of our conclusions was that the requirements are task-specific. Users' needs for access to data derive from their tasks and the responsibilities that they hold in them. This has led us to adopt a role- and task-based model for access control, enabling a wide range of coarse- and fine-grained protection policies to be defined with respect to cooperative tasks.

But the management need is for the definition and enforcement of security policies at the highest level possible. Our approach to the resolution of this conflict (i.e. that users' needs are fine-grained but management need is for coarse policy definitions) is to support the notion of security templates for generic tasks. These specify initial roles and categories of data object for a class of task. We provide editing tools that enable templates to be set up and edited. We also provide tools that enable new instances of tasks to be created with specific users designated for the roles. The access rights of users in roles are derived from the task template, but they can of course be edited (with the authority of a task manager) as the task progresses.

Further details of our protection model and its implementation can be found in [CDR98].

3. Security system design

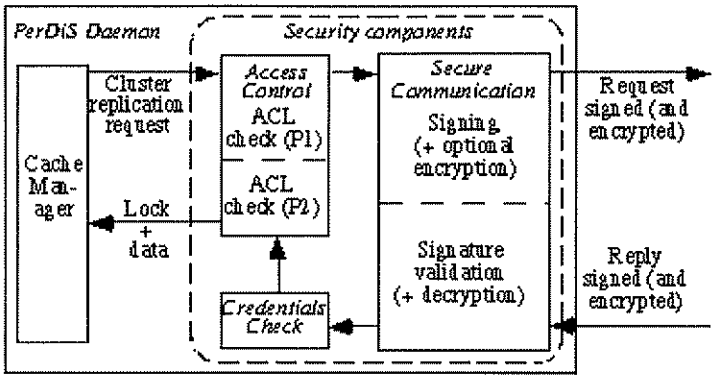

Figure 2. Security components within the PerDiS Daemon

Figure 2 illustrates the architecture of the security components of the PerDiS platform. The figure shows the security components within a single PerDiS Daemon and the interactions between them in order to generate secure requests to other PerDiS Daemons and handle the resulting replies. The same software configuration will exist at each machine that supports PerDiS.

The requests are originated by a component called the Cache Manager. This component has no security functions, it is responsible for the replication of data as required at each site. The task of the security components is to apply access control, authentication and encryption to the contents of messages as necessary to ensure that the relevant protection and secure communication policies are enforced.

Protection (i.e. the enforcement of access control, preventing users from accessing data unless they have permission to do so) is enforced whenever data is transferred between machines. Furthermore, as a consequence of the trust model discussed in Section 2.1, access control is applied at the *sending* machine when a user attempts to a read on data that is not present in the local memory and at the *receiving* machine when data is transmitted to another machine after it has been updated. These requests are rejected unless the user has the appropriate access rights.

Figure 2 shows how this is achieved when the local Cache Manager obtains data for an application running on behalf of a principal P1. After P1's rights have been checked locally a request is signed, (optionally) encrypted and despatched to the PerDiS Daemon currently holding the required data. The second PerDiS Daemon validates the request, checks P1's credentials and access rights and if they are valid returns a signed (and optionally encrypted) reply. Finally, the credentials of P2, the principal behind the second PerDiS Daemon are checked at the first PerDiS Daemon before the data is passed to the Cache Manager.

4. Conclusion

We have outlined the security design for the PerDiS groupware platform. A rationale is given for the choice of a role- and task-based access control model in a general-purpose groupware platform designed for use in a widely distributed environment (the virtual enterprise). Although some aspects of the design have been constrained by the architecture of the PerDiS platform we believe that most of the features included in our design are applicable to groupware in a wider context.

A preliminary release of the PerDiS platform is available for evaluation [PP98]. This release includes the access control code, but not secure communication. We plan to secure communication before the end of 1998.

We also hope to release the PerDiS security and access control code in the form of a toolkit for integration with other systems.

References

[CD94] G. Coulouris and J. Dollimore, Requirements for security in cooperative work: two case studies Tech. Report 671, Department of Computer Science, Queen Mary and Westfield College, May 1994. (http://www.dcs.qmw.ac.uk/research/distrib/perdis/)

[CDR98] George Coulouris, Jean Dollimore, and Marcus Roberts. Role and task-based access control in the PerDiS project. In *Workshop on Role-Based Access Control* , George Mason University, VA (USA), October 1998. ACM. (http://www.dcs.qmw.ac.uk/research/distrib/perdis/)

[HY98] Tim Hudson and Eric Young, SSLeay FAQ, June 1998, (http://www.ssleay.org/).

[LABW92] Lampson, B.W., Abadi, M., Burrows, M. and Wobber, E. Authentication in Distributed systems: Theory and Practice. *ACM Trans. on Computer Systems* , 10, 4, Nov. 1992. Pages 265-310. (http://research.microsoft.com/users/blampson/45-AuthenticationTheoryAndPractice/Abstract.html)

[NETS96] Netscape Corporation, *Secure Sockets Layer (SSL)* , v 3.0, 1996.

[PP98] PerDiS Partners,*The PerDiS Preliminary Platform* (for Unix and Windows NT), (http://www.perdis.esprit.ec.org/), 1998.

[RCD98] Marcus Roberts, George Coulouris, Jean Dollimore. Secure communication in non-uniform trust environments. In*ECOOP W. on Dist. Object Security* , Brussels (Belgium), July 1998. (http://www.dcs.qmw.ac.uk/research/distrib/perdis/)

[SCHN96] Bruce Schneier, *Applied Cryptography* , Wiley, 1996.

[WABL93] Wobber, E., Abadi, M., Burrows, M. and Lampson, B.W., Authentication in the TAOS operating system, *ACM Operating Systems Review* , December 1993. Pages 256-269. (http://research.microsoft.com/users/blampson/51-AuthInTaos/Abstract.html)

George Coulouris is Professor of Computer Systems at Queen Mary and Westfield College, University of London. He has published two books and more than 30 research papers and is a Fellow of the British Computer Society. He is currently researching architectures and platforms for synchronous groupware and distributed multimedia applications and has a special interest in protection models for groupware. He welcomes comments and queries by email to George.Coulouris@dcs.qmw.ac.uk.

Footnote:
The PerDiS Persistent Distributed Store , European Union IVth Framework Long-Term Research Project 22533. See: http://www.perdis.esprit.ec.org/ for general details or http://www.dcs.qmw.ac.uk/research/distrib/perdis/ for further details of the security aspects of the work.