SECURE ROLE BASED MESSAGING

David Chadwick[1], Graeme Lunt[2] and Gansen Zhao[1]
[1]*ISSRC, ISI, University of Salford, Salford, M5 4WT, UK;*
[2]*Nexor Limited, Bell House, Nottingham Science & Technology Park, University Boulevard, Nottingham, NG7 2RL, UK*

Abstract This paper describes a secure role based messaging system design based on the use of X.509 Attribute Certificates for holding user roles. Access to the messages is authorised by the PERMIS Privilege Management Infrastructure, a policy driven role based access control (RBAC) infrastructure, which allows the assignment of roles to be distributed between trusted issuing authorities, and allows a change of access control policy at runtime. Messages can be sent by roles and users, and can be sent to roles and users. Messages are secure in their exchange between senders and recipients. Details of the security and messaging design are presented.

Keywords: X.509, Attribute Certificates, Role based Messaging, PERMIS, PMI

1. Introduction

Messaging systems like email systems are widely used to enable communication between people. In the setting of organizations, a message is sent to a person with the assumption that the receiver is the person who is responsible for dealing with the message's contents. To some degree, it requires that senders know about the identities of people corresponding to specific duties and issues, and that they can always get up-to-date information on the relationships between issues, duties and the people who are responsible for them.

From a business's point of view, the destination of these messages is organisational roles instead of the physical occupants of roles. A role is an abstract model in an organization, which specifies a set of duties and tasks and can be associated with a set of people in the organization. People are related to the duties and tasks by being assigned as an occupant of the corresponding role. Role assignment is dynamic, and should be instantaneous, especially when a role is being removed from a currently role occupant.

Security is required when the information being carried by a message is important and/or confidential. Message security means various things including: confidential messages can be accessed only by authorised entities, including

roles and users; message contents have integrity and are protected from being modified during the course of their transportation; recipients cannot falsely deny having received messages that have been delivered; and the identity of message senders can be verified at any time after the messages have been sent.

Most existing messaging systems provide only person to person messaging and lack the ability to send messages to and from dynamically changing role occupants.

The purpose of the current research at Salford is to design, build and test a secure role based messaging system that can provide for the secure exchange of messages between organisational roles. This paper describes such a design based on the use of: X.509 attribute certificates [9] for holding user roles, PERMIS [1], a role based access control (RBAC) [4, 5] infrastructure, and user and role public/private key pairs. Access to both user mailboxes and role mailboxes is authorised by the PERMIS Privilege Management Infrastructure (PMI), an XML policy driven RBAC infrastructure, which allows the assignment of roles to be distributed between trusted issuing authorities. The design is achieved in a way that has the least impact on existing systems and standards.

The rest of this paper is structured as follows. We present the system requirements for secure role based messaging systems and the current challenges in Section 2, and propose a system design for secure role based messaging systems in Section 3. The details of secure messaging are elaborated in Section 4. We conclude with a review of the related work in Section 5 and conclusions of the system design in Section 6.

2. Requirements and Challenges

This section presents the system requirements and challenges for secure role based messaging systems.

2.1 System Requirements

A secure role based messaging system should have all the following properties:

- Privacy and confidentiality. Messages are delivered in an encrypted manner, and are accessed by only authenticated users and roles through authorised operations.
- Integrity and authenticity. Senders can sign messages, which enable recipients to verify the sender identity. Messages will not be modified during the course of their delivery.
- Time sensitivity. Role occupants are able to access the role mailbox only during the time that they officially hold the role

- Accountability. It should be possible to determine who acted in any specific role at any particular time.
- Scalability. The system should support multiple role occupants both concurrently and consecutively e.g. any current role occupant should be able to access and respond to any of the messages in the in-tray, and read all the responses sent by all the other role occupants in the out-tray.
- Manageability. Managers should be able to dynamically allocate and remove roles from people and they should then immediately be able to access role mailboxes (or not) without having to reconfigure the system
- Distributed management of roles. Different managers in different organisational units should be able to allocate and remove roles from people under their control.
- Policy based access control. Authorisation can be represented by policies, which can be set or modified to specify who is trusted to allocate which roles to which users, and which access rights are given to which roles.
- User friendliness. User friendliness is required for both use and management. It should be simple for users to access messages whilst acting in the capacity of a role. In the case that users hold a number of roles, it should also be simple to select which role users wish to exercise at any particular time. It must also be easy for managers to allocate and remove roles, and for the security officer to set the policy controlling access to the role based messaging system.

2.2 Challenges in Secure Role Based Messaging

There is significant interest in role based messaging. Such a system presents a number of challenges, especially when encryption and digital signing are involved, since keys need to be assigned to a role, even though the role could be occupied by zero, one or several real people at any given time.

Within a large and structured organisation, people often want to send a message to a given role within that organisation, rather than to a specific individual. Unfortunately most email systems are person to person and do not inherently support roles. The gap between existing messaging systems and applications is exaggerated in large organisations, which have complex organization structures and people continually adopt different roles during their professional life.

Two of the current ways of supporting role based messaging are using dedicated role mailboxes and using distribution lists. When using dedicated role mailboxes, role occupants share a role mailbox by sharing the role password or role private key to access the role mailbox. Sharing passwords brings a risk

of password disclosure, and the risk will increase when the number of role occupants increases or the change of role occupants becomes frequent. Furthermore, it is difficult to stop someone from accessing a role mailbox after the role has been removed from him or her without changing the role password. Sharing role private keys is a better option if the keys are held in hardware tokens. The tokens can then be physically removed from the role occupant when the role is removed. But if the role private key is held in an encrypted file it is little better than sharing a role password, since the file can be copied at will.

When distribution lists are used, messages sent to a role are copied to each role occupant. One of the drawbacks of this mode is that it does not cater for the temporal nature of role occupancy without significant management overhead. Removed role occupants will continue to have access to the messages that were delivered to the role prior to their removal, while new added role occupants will not be able to access those messages that were delivered to the role before their role assignment was made.

The issue is further complicated when secure messaging is required. Messages sent by a role need to be signed by a role, whilst encrypted messages received by a role need to be read by all role occupants. Role occupants could be given copies of the role private key(s), but the management overhead of these keys, both by the individual and by the administrator would be prohibitive.

Enhancing the distribution list expansion method to re-encrypt the message for each individual role occupant does not allow new role occupants to access the messages that have already been delivered, nor does it cater for role occupants whose role assignments are revoked after messages have been sent but not read and acted upon.

3. System Design

The work reported in this paper is an attempt to develop a framework for secure role based messaging, which is flexible enough to cater for different security and messaging requirements whilst having the least impact on existing systems and standards.

3.1 System Architecture

Figure 1 shows an architecture design for a secure role based messaging system. The main components of the architecture are the Message User Agent (MUA), the Message Transfer Agent (MTA), the Internet Mail Access Protocol 4 Server (IMAP4 server), and the Role Gatekeeper. The MUA is a component that provides users with facilities for sending and receiving messages. It mediates the communication between users and other components, and the communication between roles and other components. The MTA component is responsible for transporting messages. It receives messages from MUAs or

remote MTAs, and delivers messages to a remote MTA or stores them locally according to the destination of the messages. The IMAP4 server provides an interface to access electronic messages that are stored in the system. Users and roles use MUAs to retrieve messages from the IMAP4 server. Messages are transmitted between MUAs and MTAs over the SMTP protocol [10], and are transmitted between MUAs and IMAP4 servers over the IMAP [2] protocol.

The Role Gatekeeper intercepts messages between the MUA and the MTA server, and communications between the MUA and IMAP4 server. The communications between the Role Gatekeeper and the other components are secure by means of authentication and encryption of messages or links. The Role Gatekeeper is responsible for all security operations regarding roles.

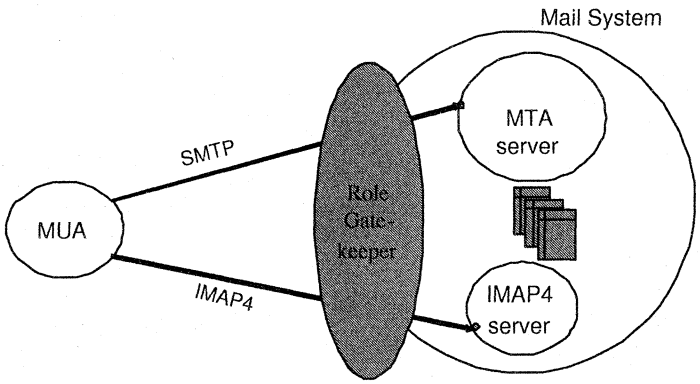

Figure 1. The Secure Role based Messaging System Architecture

3.2 The Role Gatekeeper

Details of the Role Gatekeeper design are shown in Figure 2. The Key components in the Role Gatekeeper include the authentication service component, the encryption and decryption service component, and the PERMIS authorisation service component. The authentication service component verifies the identities of message senders, and the identities of users and roles who are requesting to perform actions within the messaging system.

The encryption and decryption service component signs (and encrypts) role messages if necessary when they are being sent, and decrypts encrypted role messages when they are being retrieved. The encryption and decryption service component has access to the role private keys, so it can sign role messages on behalf of a role by using the role's private key. The encryption and decryp-

tion service component eliminates the requirement of role occupants to hold role private keys for signing and decrypting role messages.

The PERMIS authorisation service component authorises all security sensitive operations within the system, ensuring that only permitted operations on messages are conducted in regard to the identity of users and roles. The PERMIS access control decision function (ADF) can reason and decide whether an operation of a user or a role is permitted or not, in regard to a specific resource, which are mailboxes in the case of this work. PERMIS also supports the distributed management of roles.

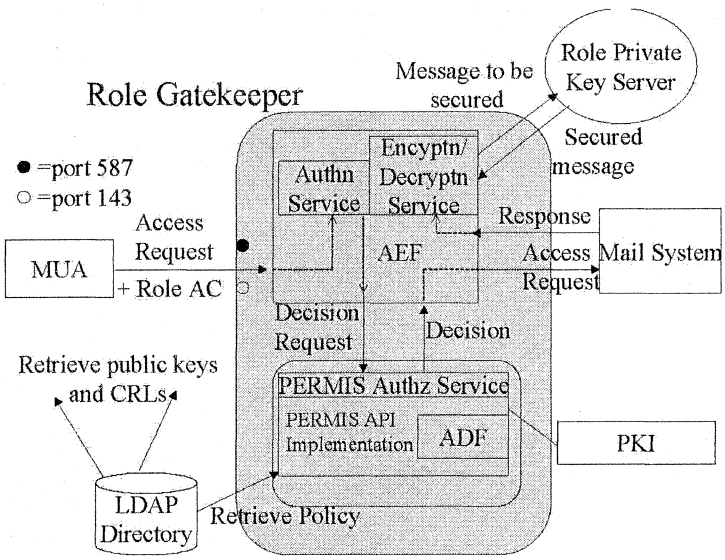

Figure 2. Role Gatekeeper

3.3 Private Keys and Pubic Keys

The proposed design assumes that each role and user is allocated his or her own public/private key pair(s). One key pair is used for digitally signing messages, named signature keys, the other for encrypting/decrypting messages, named encryption keys. The user private keys are held in a smart card, encrypted file or similar that is only accessible to them, while the role private keys are held in a role private key server. Users can access their own private keys, but role occupants have no access to the role private keys. The Role Gatekeeper will access the role private keys on behalf of role occupants when it is necessary to use the role private keys to sign or to decrypt messages. Both

the public keys of users and roles are held as X.509 public key certificates [9] in an LDAP directory [18].

3.4 Attribute Certificates

An Attribute Certificate (AC) is a data structure which contains a set of attributes for an entity, and it is required to be signed by an attribute authority (AA), which is an authority trusted by the system or the user [16]. A Role Assignment Attribute Certificate (RAC) specifies a user's assignment of a role. Role occupancy is conferred upon a user by being in possession of an X.509 RAC. Given a RAC and a trusted root AA, called the Source of Authority (SOA) in X.509, it can be verified whether the RAC is issued by a trusted AA or not, and it can also be determined whether a user is assigned to a role. A Role Specification Attribute Certificate [9] or Policy Attribute Certificate (PAC) [16] specifies the authorised operations of a role on a specific resource. In PERMIS, for efficiency reasons, all the PACs are collected together into one PERMIS authorisation Policy Attribute Certificate. A user acting in a trusted role will be authorised to perform a specific operation on a specific resource if the PERMIS policy specifies that the role is allowed to perform the operation on the resource.

4. Secure Messaging

The system framework described in Section 3 works by authenticating users and their roles and authorising their operations, i.e. sending, delivering and accessing messages.

4.1 Authentication and Authorisation

The authentication service component within the Role Gatekeeper is responsible for authenticating both the identity of a user and the identity of a role. Users hold their private keys by themselves, and make them available for the MUA. The MUA can then use the private keys to achieve authentication with the Role Gatekeeper. One of the possible ways is that the Role Gatekeeper requires the MUA to sign a specific message, and verifies the signature on the message. The general submission mechanism is specified in [7].

Role occupants do not have access to role private keys, thus they have no way to prove their role identity by directly using role private keys. In the proposed design, a user gains its authentication as a role occupant by getting its own authentication and then proving its assignment of a role through a valid X.509 RAC. In this way, when a user wants to login in as a role, he/she will first login into the system as a user as above, and then present a RAC to the system, which specifies that the user has been assigned to the role.

The PERMIS component of the Role Gatekeeper authorises operations of both users and roles. RACs and the PERMIS PAC are the most important security tokens for the PERMIS authorisation service component to authorise a user's and role occupant's operations. The authorisation requires a user or role occupant to present one or more RACs, which can show that they are allowed to perform the specific operations. Given the identity, the requested operation and the related RACs, the PERMIS Access Control Decision Function (ADF) can reason and decide whether the request of performing the operation is allowed or not, in regard to the target mailbox.

PERMIS can operate in either push or pull mode. In push mode, the user pushes their RACs to the Role Gatekeeper to present to PERMIS. In pull mode, the PERMIS component fetches the user's RACs from one or more configured LDAP directories.

The way the system works can best be described by considering in detail each of the following three scenarios:

- a user sends a secured message to a role
- a role sends a secured message to a user
- a role sends a secured message to a role

4.2 User to Role

A user (MUA in the figures) wishes to send a secured message (digitally signed and/or encrypted) to a role. In normal email systems, when encrypting a message, the user would first obtain the encryption public key of the role/recipient from the LDAP directory and also obtain the latest revocation information for the role/recipient certificate (e.g. Certificate Revocation List(CRL) from the LDAP directory or OCSP [14] response from OSCP responder) to ensure that the certificate has not been revoked. The user would then digitally sign and/or encrypt the message, and send it to the SMTP server. The order of signing and encrypting would typically be determined by the MUA software, unless it was configurable by the user. ESS triple wrapping [17, 8] states the order should be Sign/Encrypt/Sign. A signature over the clear content, as opposed to encrypted content, has much more value.

In our design, because users do not have access to role private keys, the order of securing messages is important. Digital signing must come first. Signing followed by encrypting allows the Role Gatekeeper to subsequently decrypt and then re-encrypt the message to role keys without invalidating the signature of the sender; otherwise the signature will be invalidated when the message is decrypted. Thus the actual order of events is as follows.

The user creates a message to a role recipient and selects the sign and/or recipient role encrypt features. If sign was selected the MUA signs the message

using the user's private signing key. The message is then transferred to the message submission port (587) on the Role Gatekeeper along with the optional recipient role encryption flag. The Role Gatekeeper, upon seeing the recipient role encryption flag, encrypts the message to the recipient role, using the role's public key obtained from the LDAP directory. (Revocation checking is also always carried out, but we will take this as a given and not mention it again). The S/MIME double wrapped message [6, 17, 8] (signed by user, encrypted to role) is then submitted to the SMTP server by the Role Gatekeeper.

We note that the encryption could have been done at the MUA as the recipient role's public key is in the directory and the originating role private key is not used in the encryption process. However, to maintain consistency with the other scenarios, we propose to always perform recipient role encryption in the Role Gatekeeper.

We further note that the link between the MUA and Role Gatekeeper may be encrypted or not to preserve message confidentiality. The security of this link is independent of the encryption of the message to the recipient role. If the link is encrypted this will have been negotiated at session establishment, using for example an SSL/TLS link.

The SMTP system distributes the mail until it eventually ends up in the role mailbox of an IMAP4 server. The secured message sits in the role mailbox of the IMAP4 server until a current role occupant logs in. A role occupant (user) authenticates to port 143 of the Role Gatekeeper and uses SASL [13] to identify themselves. The user then requests to access a particular mailbox using standard IMAP commands. The Role Gatekeeper determines if the user has access to the requested mailbox by determining if the user has an appropriate X.509 RAC. If no RAC is presented, the user's request is passed straight through to the mail server and he/she is only granted access to the default mailbox of their own username (INBOX in IMAP4). Such exchanges are no longer considered by this paper

If a RAC is presented, PERMIS authorises the operations according to the RAC and the related PERMIS policy. If PERMIS grants access, the Role Gatekeeper will log into the role mailbox on behalf of the user, using the role mailbox password that it holds. (Note that users do not know the passwords to the role mailboxes). When the user sends request messages to access the folders within the mailbox e.g. via the IMAP SELECT command, then the embedded folders and mail headers are returned to be displayed on the user's terminal. When the user wishes to fetch the contents of an encrypted message (e.g. via the IMAP FETCH command), then the Role Gatekeeper extracts the encrypted key information, sends it to the key server, and then attaches the response to the message before returning it to the user.

The encrypted key information $\mathrm{E}\{Km\}_{Pkr}$ contains the symmetric key used to encrypt the message, Km, encrypted with the public key of the re-

cipient role, Pkr. The key server, which has the private keys of all the roles, is able to decrypt the encrypted key information using the role private key of the role encryption key pair, Prr. It then re-encrypts the symmetric key, Km, using the user's public encryption key, Pku, which it can obtain from the LDAP directory. Note that we are currently working on the design of a policy controlled email system, which will allow some role occupants not to have access to some policy labelled messages, but description of this feature is out of the scope of this paper.

When the role occupant receives the encrypted message, (s)he is able to decrypt the message using their own private decryption key, and then validate the signature of the sender.

4.3 Role to User

A user logs into the IMAP4 server as before, by sending their RAC to the Role Gatekeeper. The Role Gatekeeper checks the validity of the RAC and if OK, allows the role occupant (the user) to download the contents of the role mailbox. The role occupant may now either reply to a message in the inbox, or create a new message to someone, acting in their official role. Once the message has been created, the role occupant selects the sign, role sign and/or encrypt functions, and if sign is selected, the MUA digitally signs the message using the role occupant's own personal private key of the signature key pair. This functionality provides a complete audit trail of which role occupant actually acted in the role at the time the message was signed. The MUA sends the message to the SMTP server via the Role Gatekeeper along with indicators specifying whether the message requires role signature and/or encryption. If this is a new connection between the MUA and the SMTP server (actually to port 587 on the Role Gatekeeper), then the user's RAC is transferred during the connection establishment phase. The Role Gatekeeper validates the RAC of the role occupant (once per session) and if the message is flagged to be role signed, it is sent to the key server for digital signing using the private key of the role signature key pair. If the message is also flagged to be encrypted, then it is encrypted to the recipient's public encryption key by the Role Gatekeeper. Finally it is submitted to the SMTP server. The resulting message may be S/MIME tripled wrapped (signed by sender, signed by role, encrypted to recipient).

When the user downloads the message from the IMAP4 server (in this case the Role Gatekeeper does not interfere with the message exchanges) the user is able to decrypt the message using the private key of their encryption key pair, then validate the signatures of the sender. (We assume here that the MUA is capable of downloading certificates and CRLs from an appropriately configured LDAP directory, and is capable of deciphering doubly signed messages).

In an alternative design that we are also building, the user includes his RAC in the message before signing the message. This binds the user to the role they are signing on behalf of. The advantage of this design is that the message is only doubly wrapped as a maximum (signed by sender and encrypted to recipient) instead of triply wrapped. Further, message encryption can be done by the MUA which relieves the burden on the Role Gatekeeper. The disadvantage is that the message is not actually signed by the sending role, and the recipient has to view the attached RAC to see that it was sent by a role.

4.4 Role to Role

This scenario is obviously a combination of the previous two scenarios. A role occupant logs into the IMAP4 server via the Role Gatekeeper, by passing a RAC at authentication time. If the role is valid, the Role Gatekeeper allows the role occupant to download the contents of the role mailbox. Any encrypted messages are passed by the Role Gatekeeper to the key server so that the symmetric encryption key can be encrypted to the public key of the role occupant's encryption key pair. The role occupant is thus able to read all messages that were encrypted to the public key of the role encryption key pair.

Any messages that the role occupant submits to the SMTP server via the Role Gatekeeper are firstly passed to the key server for digitally signing by the private key of the role signature key pair, and then they are encrypted to the public key of the recipient role's encryption key pair, resulting in an S/MIME triply wrapped message (or alternatively the sender includes his RAC in the message, signs the message and then encrypts it for the recipient role).

5. Related Work

Mont et al [12] describe a role based secure messaging service used in a health care setting. The service employs Identifier Based Encryption to protect messages. Senders decide the permitted role(s) who can view the message, and the messages will be encrypted with a string describing the permitted role(s). A recipient has to be authenticated as a member of at least one of the selected roles by the trust authority before getting a decryption key for the message. This work requires all users to be assigned a role before they can interact with the system, which is not practical to some degree.

Microsoft [11] released Microsoft Windows Server 2003 with a Rights Management System (RMS) that enables enterprises to add security information to files produced using Microsoft Office 2003 applications. The added security allows an author to limit the circulation and operations of a document. A header containing the security control policy is added to the file. The system also provides facilities for administrators to generate templates to define access control policies. One of the drawbacks is that RMS is provided without a

mechanism to specify access control policies for groups and roles. Some may argue that Microsoft Active Directory can be integrated with the system and provide mechanism for controlling group permissions. For users external to the enterprise, Microsoft mandates the use of the Passport authentication service, which is a service provided by Microsoft, to allow these users to produce licenses for their files. However, it is not yet clear how the interface between external users and the enterprise is managed and there is no provision for binding users to roles.

MailRecall^{TM} is produced by Authentica [3]. It provides plug-ins for several popular email clients with the ability of keeping e-mail's privacy and protecting emails from unauthorised users, even after delivery. MailRecall^{TM} uses content security policies to determine the expiration of messages and authorize operations on emails. These policies can be configured individually by users or centrally, in accordance with corporate policy. When a message is sent outside the organisation the external recipient can be automatically registered and a browser plug-in is downloaded when the message is opened. The plug-in allows the recipient to view the protected message. Furthermore the web viewer can be configured to prompt the recipient to install the email client plug-in. Although MailRecall^{TM} provides several security control features, it fails to provide facilities to define a security policy at the group level or from a role's perspective.

The Omniva Policy Manager package [15] offers functions that are similar to MailRecall^{TM} , and it is available as a plug-in for Microsoft Outlook. It does provide a means of applying policies to groups of users, using existing directories and external recipients can read, but not directly respond to messages, using a web browser. However, no provision is made for addressing mail to role mailboxes.

6. Conclusions

This work presents a design for a secure role based messaging system, which is based on X.509 role assignment attribute certificates and the PERMIS policy driven role based authorization system. The proposed design has been developed with the effort of making the least number of modifications to the existing Email systems and protocol standards. The assumption is that such a design will facilitate its deployment within enterprises.

We have two variations on the design for sending digitally signed role based messages. We are currently building both systems and will report on the implementation, performance and usability in due course.

Acknowledgments

The authors would like to thank Nexor Ltd who is sponsoring this research.

References

[1] D. Chadwick, A. Otenko, and E. Ball. Role-based access control with X.509 attribute certificates. *IEEE Internet Computing*, pages 62–69, March-April 2003.

[2] M. Crispin. RFC 3501: Internet Message Access Protocol - Version 4rev1. Request For Comment, Network Working Group, March 2003.

[3] Victor DeMarines. MailRecall: Secure E-mail for the Enterprise, May 2004. Authentica, Inc.

[4] David Ferraiolo and Richard Kuhn. Role-based Access Control. In *Proceedings of 15th National Computer Security Conference*, 1992.

[5] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST Standard for Role-based Access Control. *ACM Transactions on Information and System Security*, 4(3):224–274, 2001.

[6] N. Freed and N. Borenstein. RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Request For Comment, Network Working Group, November 1996.

[7] R. Gellens and J. Klensin. RFC 2476 - Message Submission. Request For Comment, Network Working Group, December 1998.

[8] P. Hoffman. RFC 2634: Enhanced Security Services for S/MIME. Request For Comment, Network Working Group, June 1999.

[9] ITU-T. Recommendation X.509, ISO/IEC 9594-8. Information Technology – Open Systems Interconnection – The Directory: Public-key and Attribute Certificate Frameworks, 4th ed., 2000. ITU.

[10] J. Klensin. RFC 2821 - Simple Mail Transfer Protocol. Request For Comment, Network Working Group, April 2001.

[11] Microsoft Corporation. Technical Overview of Windows Rights Management Services for Windows Server 2003, November 2003. Microsoft Corporation.

[12] M.C. Mont, P. Bramhall, and K. Harrison. A Flexible Role-based Secure Messaging Service: Exploiting IBE Technology for Privacy in Health Care. In *Proceeding of the 14th International Workshop on Database and Expert System Applications*. IEEE, 2003.

[13] J. Myers. RFC 2222: Simple Authentication and Security Layer (SASL). Request For Comment, Network Working Group, October 1997.

[14] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. RFC 2560 - X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol - OCSP. Request For Comment, Network Working Group, June 1999.

[15] Omniva Policy Systems. Omniva Policy Manager Technical White Paper, January 2004. Omniva Policy Systems.

[16] Rolf Oppliger, Günther Pernul, and Christine Strauss. Using attribute certificates to implement role-based authorization and access controls. In S. Teufel K. Bauknecht, editor, *Sicherheit in Informationssystemen (SIS 2000)*, pages 169–184, Zurich, 2000.

[17] B. Ramsdell. RFC 2633: S/MIME Version 3 Message Specification. Request For Comment, Network Working Group, June 1999.

[18] M. Wahl. RFC 2251 - Lightweight Directory Access Protocol (v3). Request For Comment, Network Working Group, December 1997.