Evolving Messaging Systems for Secure Role Based Messaging

Gansen Zhao
Computing Laboratory
University of Kent
Canterbury, CT2 7NF
United Kingdom
gz7@kent.ac.uk

David W Chadwick
Computing Laboratory
University of Kent
Canterbury, CT2 7NF
United Kingdom
d.w.chadwick@kent.ac.uk

Abstract

This paper articulates a system design for the secure role based messaging model built based on existing messaging systems, public key infrastructures, and a privilege management infrastructure, which enables role-oriented secure communication. Users can send and access messages on behalf of a role. Access to the messages is authorised dynamically according to the authorisation policies conveyed by X.509 Attribute Certificates. The architecture design extends the current messaging systems without invalidating the system's compliance with existing standards, and enables easy integration with existing messaging systems. This paper also contributes to providing security features based on architecture design, and demonstrates the deliberative architecture design for information confidentiality and privacy.

1. Introduction

Messaging systems like email systems are widely used to enable communication between people. In the setting of organizations, a message is usually sent to a person with the assumption that the receiver is a member of a specific role and is responsible for dealing with the message in that capacity.

A Role is a position or function of an organisation, associated with zero or multiple people who are usually authorised to perform certain transactions. 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 current 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, such as roles and users; message contents are inviolate and are protected from being modified during the course of their transportation; senders cannot falsely deny having sent messages that have been delivered; and the identity of message senders can be verified at any time after the messages have been sent.

The purpose of the current research is to design, build and test a secure role based messaging system that can provide for the secure exchange of messages between organisational roles. With secure role based messaging systems, role occupants can send and access messages on behalf of roles, all operations performed by role occupants are authorised, and communications are secure with the security features of non-repudiation, authenticity, confidentiality, integrity, and distributed security management.

The aim of this paper is to address the software engineering issues related to the development of architecture for the secure role based messaging model proposed in [1]. More specifically, this paper advances the state of the art in secure role based messaging in the following ways. Firstly, it presents a construction model that enables the extension of current messaging systems without invalidating their compliance to existing standards. Secondly, it implements the proposed secure role based messaging model incrementally, so that either senders or receivers or both may implement the model independently. More specifically, the paper demonstrates how to extend current messaging systems to achieve secure role based messaging while keeping compliance with existing standards and enabling easy integration with existing systems

This paper is organised as follows. In section 2 we present briefly the proposed secure role based model and the challenges of implementing the model in today's messaging environment. Section 3 presents the system design, which evolves the existing messaging system architecture implemented in most messaging systems today, to achieve the secure role based model. Section 4 reviews related work and contrasts it to the current research. Section 5 provides a conclusion of this work and section 6 presents a vision of the future work that is needed and its challenges.

2. Secure Role Based Messaging Model and Challenges

We have proposed a secure role based messaging model [1], which is based on the use of: X.509 attribute certificates [11] for holding user roles, the PERMIS Privilege Management Infrastructure (PMI) [2] for access authorisation, and user and role public/private key pairs. Access to both user mailboxes and role mailboxes is authorised by the PERMIS PMI, which is a role based access control (RBAC) [5, 6] infrastructure. The proposed secure role based messaging model enables secure communication between roles. Messages can be sent by a role and be sent to a role, without considering the identity of the physical person who deals with the messages. Communications satisfy most of the security requirements presented earlier [1].

2.1. Secure Role Based Messaging Model

The proposed secure role based messaging (RBM) model is based on both Public Key Infrastructure (PKI) and Privilege Management Infrastructure (PMI) models. Key pairs are allocated to each role and each user. The public keys of users and roles are held as X.509 public key certificates [11] in an LDAP directory [21].

X.509 Attribute Certificates (ACs) are used to convey authorisation related information from a trusted source [18]. This includes both the Role Assignment Attribute Certificates (RACs) and Policy Attribute Certificates (PACs). All the authorisation decisions are made by the PERMIS policy decision point (PDP) [20] according to the authorisation information.

Secure role based messaging enables role occupants to send and receive message on behalf of roles without sharing passwords or keys. A trusted entity is responsible for: managing the role private keys, signing messages, and decrypting and encrypting a message's encryption keys on behalf of roles using role private keys. The trusted entity is not given access to the encrypted messages. Conversely, the entity that has access to the encrypted messages never has access to a decryption key. The limitation of the access of different components provides better security as discussed in section 3. Role occupants operate on role messages by being authenticated with their own individual identity and all operations are subject to authorisation before execution. A role occupant's operation, which is performed on behalf of a role, will be transformed into an operation originating from the role. Messages are reconstructed to enable secure communications on behalf of roles, by adding role signatures, encrypting messages to roles, and rewrapping the encrypted messages.

2.2. Challenges

The challenges of developing and implementing the proposed secure role based model are threefold.

1. **To maintain standards compliance when providing the extended role based services**. The model provides extension to existing messaging systems, while the existing mail systems are constrained by various standards, including SMTP [13], POP3 [16], IMAP4 [3], MIME [7], S/MIME [19], Message Submission [9], and other related standards. Messaging systems are not standalone systems. Communication and interoperability with other systems are of paramount importance. Non-compliance with the existing standards will not only reduce the interoperability between the extended RBM system and existing systems, but in fact will seriously hinder the deployment of the extended RBM system.

2. **To achieve integration of existing systems when providing extended role based services.** Many millions of messaging systems exist in the existing messaging infrastructure, most of which only provide basic person to person messaging. It is sensible, from the perspectives of both development cost and deployment cost, to extend the existing systems to provide secure role based messaging, instead of replacing them. Further, in some cases, it may be impossible to replace the existing systems because the cost of purchasing and reconfiguring the whole system is unaffordable, or it is impossible to smoothly switch the existing system to a new one because of the involvement of numerous users and the difficulty of migrating data from an old system to a new one.

3. **To design security into the system architecture.** The evolution of existing systems may introduce new security threats. These newly introduced threats could compromise the new security features and functionalities that are being added. The architecture of the target system should be built in such as way that security is designed into the system. It is often very difficult or impossible to bolt on security afterwards.

Conflicts exist between maintaining standards compliance and extending services due to the limited operations defined by the existing standards, and it is also difficult to design the extension model that is generic enough to be applied to extend most of the existing messaging systems. The extension and evolution of existing messaging systems should be achieved with system protections, preventing the implemented systems from being compromised by newly introduced threats. Special design techniques are required to tackle the above challenges when we develop and implement the proposed secure role based messaging model.

3. Evolving Existing Systems for Secure Role Based Messaging

In general, a messaging system contains user agents to enable users to compose, send and receive messages, a message submission agent to validate the user's submissions, a message delivery agent to transmit messages and a message store to store and provide message access for users.

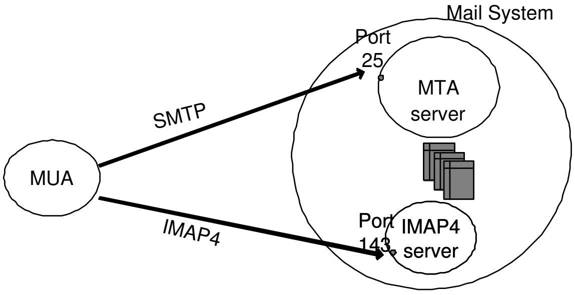

Figure 1. The Generic Architecture for Messaging Systems

Figure 1 shows a generic architecture for messaging systems. The Message User Agent (MUA) is the component that provides users with facilities to compose, send and receive messages. It mediates the communication between users and the other components. The Message Submission Agent (MSA) receives composed messages from MUAs, validates them, inserts compulsory or missing fields, makes necessary modification of messages and then forwards them to the Message Transfer Agent (MTA). The MTA is responsible for transferring messages. It receives messages from a local MUA and/or MSA or a remote MTA in other systems, delivers messages to other systems if the destination of the messages is not a local address, or just hands them over to the Message Store (MS) if the destination is a local address. Two message store protocols are standardised for use on the Internet - the Post Office Protocol Version 3 (POP3) and the Internet Message Access Protocol (edition 4) (IMAP4). IMAP4 is a more sophisticated protocol than POP3, and only IMAP4 will be considered in the rest of this paper (although the same principles can be applied to POP3). The MS stores and manages messages, and provides an interface for users to access and manage messages over the IMAP4 and POP3 protocols.

Existing messaging protocols can be fitted into this architecture as follows. IMAP4 can be placed between MUAs and Message Stores (MSs), SMTP can be placed between MUAs and MSAs, and MSAs and MTAs. Both MIME and S/MIME can be used to format messages. Thus, users can send messages over SMTP and access and manage messages over IMAP4 in both the situation of messaging on behalf of themselves or on behalf of roles.

3.1. System Architecture for Secure Role Based Messaging

Figure 2 shows the architecture design for a secure role based messaging system. The architecture design introduces to the generic architecture the Role Gatekeeper to mediate the communication between users and the message servers. The Role Gatekeeper implements a proxy pattern [8]. It interacts with the MUA in the same way as a MSA and a MS, and interacts with the MSA and MS in the same way as a MUA. Thus the Role Gatekeeper enables the MUA to send and access role based messages in the same manner and using the same standards as it does today to access interpersonal messages.

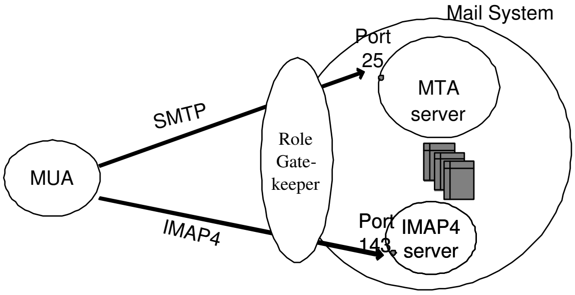

Figure 2. The Secure Role based Messaging System Architecture

The Role Gatekeeper intercepts messages between the MUA and the MSA server, and between the MUA and MS. The responsibilities of the Role Gatekeeper are as follows.

1. **Impose role based access controls.** Users are authenticated before they can perform any role related operations, and all the operations must be authorised. The Role Gatekeeper intercepts all the operations, and imposes access controls on the users according to the prevailing RBAC policy and the user's current role(s). Unauthorised operations will be prohibited by the Role Gatekeeper.

2. **Manage role mailboxes.** The Role Gatekeeper extends the message management services provided by existing messaging systems, providing them with role mailbox management abilities. The extension of services is achieved by modifying messaging commands submitted by MUAs to MSs and MSAs. The modified commands are legal commands as defined by the standards of SMTP and IMAP4.

3. **Provide security features for role based messaging.** Two kinds of role based messaging security features are provided: signing messages submitted by a role occupant on behalf of a role, and allowing a role occupant to decrypt a message sent to his current role which was previously encrypted to the role only.

3.2. The Role Gatekeeper

The Role Gatekeeper acts as a proxy intermediately between the MUA and the message servers (MSA and MS). The proxy style design of the Role Gatekeeper ensures that the target systems remain compliant with the existing standards and that the Role Gatekeeper is transparent to both the MUA and the message servers.

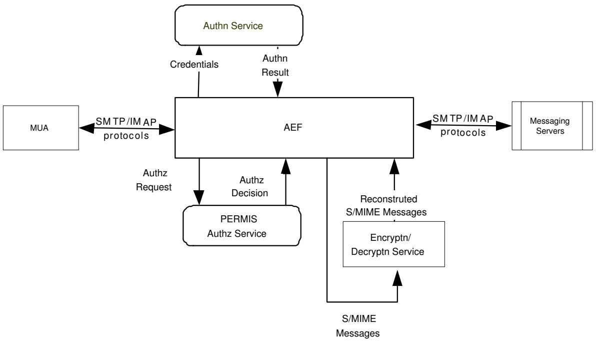

Figure 3. Design of the Role Gatekeeper

Details of the Role Gatekeeper design are shown in Figure 3. This is based on the concepts of the ISO Access Control Framework standards [12]. The key components in the Role Gatekeeper include the Access control Enforcement Function (AEF) [12] (which is also called a Policy Enforcement Point (PEP) [20]), the Authentication Service, the Access control Decision Function (ADF) [12] (which is also called a Policy Decision Point (PDP) [20]), and the Private Key Handling Service (PKHS). In our implementation, the PERMIS RBAC system is used as the ADF/PDP.

The Access control Enforcement Function (AEF) intercepts the SMTP/IMAP communication between the MUA and the messaging systems, and is in overall control of the communications. It enforces decisions made by the ADF/PDP and the authentication service.

The Authentication Service verifies the identities of users who are requesting to perform actions within the messaging system.

The Access control Decision Function (ADF) is responsible for authorising all security sensitive operations within the system. The authorisation decision is made according to the current policies.

The Private Key Handling Service (PKHS) is responsible for undertaking all actions involving role private keys. These include: signing messages on behalf of a role and decrypting symmetric encryption keys encrypted for roles. The Private Key Handling Service is the only component that has access to the role private keys.

3.3. The Access Control Enforcement Function

The AEF is designed based on the Decorator pattern [8], which allows the system to perform extra operations or add additional responsibilities to extend the functionality of the original operations submitted by the MUAs. The AEF also imposes access control to the original operations and maps user operations to role operations when a role occupant is operating on behalf of a role.

User identities are authenticated prior to any operation. All the operations are directed to the AEF and will be authorised before the request is actually sent to the message servers. Authentication is carried out by the Authentication Service, which could be an external service such as an LDAP directory holding usernames and passwords, or a PKI. Authorisation decisions are made by the ADF, according to the user's roles, the requested action and the current policy.

When a user issues a request to operate on a message of a role, the request will be passed to the message servers with the role identity, providing the user is authorised to do that by the ADF. Extra requests may be sent to the message servers along with the original request, or a modified request may be sent, depending on the request.

The corresponding response(s) from the message servers will be returned to the AEF in regard to the role's identity. The AEF will reconstruct the response(s) to the MUAs using legal SMTP or IMAP4 response(s) according to the original operation issued by the user.

Further, the AEF is also responsible for reformulating messages when a role occupant accesses messages sent to a role. Secure messages sent to the role may be encrypted to the role using the role's public key. In this case, only those who have access to the role private key can decrypt and read the messages. To enable role occupants to read messages on behalf of roles without the need to deliver the role private key to all the role occupants, the AEF will reformulate the encrypted messages. The encrypted symmetric key of the message will be extracted from the original message, and then be passed to the Private Key Handling Services for conversion (as discussed in section 3.4). The re-encrypted symmetric key will then supplement the encrypted symmetric key in the original encrypted message, thus the current role occupant and future role occupants (through a repeat of this process) are able to read the message without directly using the role private key.

In this way, the AEF communicates with MUAs and message servers over SMTP and IMAP4, which are the current standards of Internet based messaging systems. The AEF introduces an authorisation service to the messaging systems by acting as an execution monitor which intercepts all the operation requests and terminates unauthorised ones. It introduces a role mailbox management service by modifying the SMTP and IMAP4 operations. These extensions of the system have no effect on the standard compliance of the original messaging systems since all message formats and protocols are confined to standard ones.

3.4. The Private Key Handling Service

This service is responsible for the management of role private keys, for generating role signatures, and for indirectly decrypting messages to roles. In other words, it is responsible for all the role private key related processing. When a role occupant wishes to send out a digitally signed message on behalf of a role, the AEF will be responsible for imposing the role signature over the message. The Hash value of the message is computed by the AEF and passed to the Private Key Handling Service. The Private Key Handling Service will transform the hash value into the role's digital signature by using the role's private key, which will be attached to the message by the AEF.

When a role occupant wishes to access an encrypted message on behalf of a role, the AEF needs to reformulate the encrypted message to make it accessible to the specific role occupant. The AEF will extract the encrypted symmetric key from the encrypted message, and hand it over to the Private Key Handling Service. The Private Key Handling Service will transform the encrypted symmetric key in the following way: firstly, it will decrypt the encrypted symmetric key using the role's private key. Secondly, the symmetric key will be encrypted for the role occupant using the latter's public key. The re-encrypted symmetric key will then be returned to the AEF for adding to the original encrypted symmetric key in the message. Thus the message will be accessible to both the current role occupant (by using his/her own private key) and any subsequent role occupant (by a repeat of this process).

To be specific, the encrypted S/MIME message is encrypted using symmetric cryptography with an encryption key K_s. The encryption key is then encrypted using the public key K_{ap} of an asymmetric key pair $< K_{ar}, K_{ap} >$ forming K_{es}. K_{es} is attached to the encrypted message. In the reconstruction process when a role occupant wishes to receive an encrypted message on behalf of a role, the encrypted encryption key K_{es} will be decrypted by using the role private key K_{ar} of the asymmetric key pair $< K_{ar}, K_{ap} >$ to yield the clear key K_s. K_s is then re-encrypted by using the public key K'_{ap} of the user's asymmetric key pair $< K'_{ar}, K'_{ap} >$. The resulting re-encryption of the encryption key K_s is K'_{es}, which is only able to be decrypted by using the private key K'_{ar} of the user's asymmetric key pair $< K'_{ar}, K'_{ap} >$.

Implicit in this process is that the sender's signature is in an inner S/MIME encoding. In this way, the reconstruction of the message does not invalidate the sender's signature. This model also ensures message security, since the Private Key Handling Service only has access to the symmetric key and not the encrypted message, whilst the AEF only has access to the encrypted message and never has access to the clear symmetric key. This separation of processing duties ensures that confidential information contained in a message will not be disclosed during the process of reformulation, unless the Private Key Handling Service and the AEF collude together. The same design is also adopted in Hasselbach's model [10] which is referred to as the design concept of "eparation of duty".

The Private Key Handling Service is based on the Filter Pattern design concept, which takes a standard input and generates a standard output. Two filters comprise the Private Key Handling Service. One is implemented to generate digital signatures for a role, and the other is implemented to re-encrypt encrypted symmetric keys for a specific role occupant. The two filters perform corresponding transformations on their inputs, and provide very simple interfaces for interacting with the other components. Their simple formats and standardized inputs and outputs enable them to be adjusted or replaced to extend or change the functionalities of the Private Key Handling Service. By choosing different filters a system can provide customised and even dynamic mechanisms for users to employ different encryption algorithms.

4. Related Work

Mont et al [15] describe a role based secure messaging service used in health care settings. Their 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 trusted central authority before getting a decryption key for the message. This work requires all users to be assigned a role by the central authority before they can interact with the system. This is neither scalable nor manageable in open worlds which have users dynamically added into the system in a distributed manner. Furthermore, the central authority has the ability to decrypt all the messages sent by the system, which compromises its security. These weaknesses mean that the system is not scalable or secure enough for global Internet based messaging.

Microsoft [14] 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 a mechanism for controlling group permissions. For users external to the enterprise, Microsoft mandates the use of its Passport authentication service 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. Furthermore, many people do not like Microsoft's Passport technology since it places Microsoft in the position of a centrally trusted authority to hold user credentials.

MailRecall^{TM} is produced by Authentica [4]. It provides plug-ins for several popular email clients, and has the ability to keep e-mails private and protect them from unauthorised users, even after delivery. MailRecall^{TM} uses content security policies to determine the expiration of messages and authorize operations on them. 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 [17] 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. 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.

Jens Hasselbach [10] presents a design for secure mailing lists which employs the re-wrapping of S/MIME messages sent to a mailing list and redistributes the re-wrapped S/MIME messages to members of the mailing list. The most important concept in this design is the separation of duty in the architecture, which requires two separate and independent components to be responsible for reformulating the messages and for processing the symmetric keys. This design makes it impossible to reveal the text contained in a message solely by either of the two components. However, because the design is based on mailing lists, it suffers from the temporal limitations that role occupancy is determined when messages are sent, not when they are retrieved. Furthermore, all role occupants receive each message, rather than the one who will act on it, and none of the others see the responses unless this is copied to the list as well.

Wolthusen [22] presents an approach for mandatory and distributed security enforcement by intercepting all the network traffic at the operating system level. Network traffic is modified at the operating system level to provide security features, including automatically changing the outgoing messages to S/MIME messages with signature and encryption, and converting incoming S/MIME messages to plain text. This approach requires no central mechanisms to support the enforcement of security features, but it does not provide role based messaging. Users are not provided with options to decide whether the message should be delivered with security features or not, and its tight coupling with the operation systems also limits its applications and reduces its ability of running in heterogeneous environments.

5. Conclusions

This paper presents a system design for secure role based messaging that can provide for the secure exchange of messages between organisational roles. It contributes to the design and development related to software engineering issues that help to build the secure role based messaging model by evolving existing systems and protecting the target systems by special system architect design. The design demonstrates how to make use of software patterns to build systems with various requirements. The design extends current messaging systems without invalidating those systems' compliance with existing standards, and achieves easy integration with existing systems.

Further, the modular organisation of the design makes the implementation flexible enough to cater for future changes. The proxy pattern adopted by the design of the Role Gatekeeper contributes to enabling the Role Gatekeeper to run transparently between both the MUAs and the Message Servers, thus providing compliance with existing standards. It also contributes to providing the Role Gatekeeper with an easy way to integrate with existing generic Messaging Servers. The AEF, being implemented with the Decorator Pattern, can be easily extended to accommodate protocols other than SMTP and IMAP4, and extended to provide more services over SMTP and IMAP4. The Private Key Handling Service based on the Filter Pattern can also be easily adjusted to provide other S/MIME compliant message wrapping mechanisms.

It should also be noticed that the design deliberately separates the duties of processing the encrypted messages and processing the encrypted encryption key. The design can ensure the privacy and confidentiality of the messages by eliminating the possibility that one of the components may be able to decrypt the encrypted messages. This deliberate design results in the AEF being responsible for processing the encrypted messages and the PKHS being responsible for processing the encrypted encryption key. It is impossible to decrypt the messages in our system without the collusion of the AEF and the PKHS.

We believe our design of the secure role based messaging model has tackled the presented challenges successfully, which extends the existing messaging standard's services, enables easy integration with existing messaging systems while keeping the systems' standard compliance, and provides architecture-based information confidentiality for the system.

6. Future Work

Future work involves implementing the design reported in this paper and augmenting messages with security policies. The augmentation of security policies with message aims at providing security control over distributed and heterogeneous environments. Messages will be protected as critical resources and all accesses to messages will be governed by the augmented security policies. This approach is expected to provide a distributed security mechanism for communication from the perspective of resource management. Challenges include building trust relations between systems, representing security policies, interpreting security policies, and enforcing security policies.

As the target domain is an open, distributed and heterogeneous environment, no assumption can be made about the relationship between individual systems, or about the behaviour of different systems. Thus it is especially important to address the issues of building trust relationships between systems and providing a consistent mechanism to interpret and enforce security policies. To build trust relationships between systems and to enable consistent interpretation and enforcement of security policies are of high priority in the research schedule.

Other challenges include how to represent security policies and how to augment messages with security policies, and so on.

Future research results will be reported in due course.

Acknowledgements

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

References

[1] D. Chadwick, G. Lunt, and G. Zhao. Secure Role based Messaging. In *Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2004)*, pages 303–316, Windermere, UK, September 2004.

[2] 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.

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

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

[5] D. Ferraiolo and R. Kuhn. Role-based Access Control. In *Proceedings of 15th National Computer Security Conference*, 1992.

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

[7] 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.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

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

[10] J. Hasselbach. A Practice-Oriented Approach to Security Enhanced Mailing Lists. In *The Sixth International Conference on Electronic Commerce Research*, Dallas, US, October 2003.

[11] 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.

[12] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3. Security Frameworks for open systems: Access control framework, 1995.

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

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

[15] M. 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.

[16] J. Myers. RFC 1939 - Post Office Protocol - Version 3. Request For Comment, Network Working Group, May 1996.

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

[18] R. Oppliger, G. Pernul, and C. Strauss. Using attribute certificates to implement role-based authorization and access controls. In S. T. K. Bauknecht, editor, *Sicherheit in Informationssystemen (SIS 2000)*, pages 169–184, Zurich, 2000.

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

[20] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. RFC 2904 - AAA Authorization Framework. Request For Comment, Network Working Group, August 2000.

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

[22] S. D. Wolthusen. A Distributed Multipurpose Mail Guard. In *The 2003 IEEE Workshop on Information Assurance*, West Point, NY, US, June 2003.