Distributed Key Management for Secure Role based Messaging

Gansen Zhao, Sassa Otenko, and David Chadwick
{gz7,o.otenko,d.w.chadwick}@kent.ac.uk
The Computing Laboratory, University of Kent
Canterbury, United Kingdom

Abstract

Secure Role Based Messaging (SRBM) augments messaging systems with role oriented communication in a secure manner. Role occupants can sign and decrypt messages on behalf of roles. This paper identifies the requirements of SRBM and recognises the need for: distributed key shares, fast membership revocation, mandatory security controls and detection of identity spoofing. A shared RSA scheme is constructed. RSA keys are shared and distributed to role occupants and role gate keepers. Role occupants and role gate keepers must cooperate together to use the key shares to sign and decrypt the messages. Role occupant signatures can be verified by an audit service. A SRBM system architecture is developed to show the security related performance of the proposed scheme, which also demonstrates the implementation of fast membership revocation, mandatory security control and prevention of spoofing. It is shown that the proposed scheme has successfully coupled distributed security with mandatory security controls to realize secure role based messaging.

1. Introduction

The Secure Role based Messaging (SRBM) model [4] provides a model for role oriented communication. Role occupants share role identities and are authorised to sign and decrypt messages on behalf of roles.

The SRBM model requires role signatures and optional encryption over all messages, which raises the challenge of sharing the private keys of the roles. SRBM systems have various implementation models. Some systems may directly share role private keys among all role occupants but this model suffers from the difficulty of key revocation. In central wrapping systems a trusted entity is employed to manage all the role private keys and key uses, but the trusted entity will be subject to attacks and forms a single point of failure [4]. Group signature and threshold cryptography systems promote distributed key sharing but they lack mechanisms to impose mandatory security controls.

SRBM system proposed here extracts the particular advantages of the above models whilst avoiding their disadvantages. Our system shares role private keys between role occupants and a trusted role gate keeper. Key shares are used cooperatively by role occupants and the role gate keeper. It also implements fast membership revocation and supports mandatory security audit and controls. We propose a distributed RSA scheme for our SRBM architecture.

The organisation of this paper is as follows. Section 2 presents a secure role based messaging scenario and the challenges it presents. Section 3 describes our distributed RSA scheme Section 4 describes the architecture of our SRBM system. Section 5 performs a security analysis on the proposed scheme. Section 6 compares our SRBM system to previous research and the conclusions are presented in Section 7.

2. A Scenario and the Challenges

The following is a hypothesised scenario for a SRBM system. Alice and Bob are both sales people in ABC Ltd, and the Sales Gate Keeper is a mail gateway for the Sales department. Alice, Bob and the Sales Gate Keeper each hold a share of the private key of the role Sales. Alice and Bob can send digitally signed messages and receive encrypted messages on behalf of the role Sales but the signing and decryption only succeeds when they are performed with the help of the Sales Gate Keeper. Collusion between Alice and Bob will not reveal any information about the Sales' role private key, and will not help them to sign or decrypt role based messages. Similarly the Sales Gate Keeper can not sign or decrypt role based messages without the participation of either Alice or Bob. Finally, the Sales Gate Keeper is not able to produce any plain text information when it helps Alice and Bob to decrypt role based messages.

The Sales Gate Keeper can audit all communications and

[1] The authors would like to thanks Nexor who is supporting the research.

impose security controls such as checking if attachments are classified or if recipients have the proper authority to read a message. The Sales Gate Keeper will only cooperate to sign or decrypt messages that are allowed by the messaging policies and the audit policies.

To revoke a role membership, the system can simply inform the Sales Gate Keeper of the revocation, and the Sales Gate Keeper will no longer cooperate with the revoked member. The membership revocation is transparent to the other parties, and there is no need to change the role Sales' public key or private key.

To implement the above scenario, a key management scheme must be developed which allows the private keys to be shared among members and a trusted entity (the role gate keeper). Collusion amongst members will not reveal the private keys. Members can use their key shares to sign or decrypt messages on behalf of their roles only when they succeed in acquiring the help of the trusted entity.

The trusted entity must be designed in a way that breaking the trusted entity on its own must not threaten the security of the whole system, and there should be a way for secure recovery. Whilst the trusted entity is involved in message decryption, the plaintext should not be revealed by it.

For accountability purposes, a signature produced by a role occupant using its key share should be able to be verified against its identity, thereby authenticating the signer. Role occupants should not be allowed to spoof their identities when performing their role based duties.

A mandatory security control is required to enforce local security policies and message based policies. All communications must be subject to the security control before the messages are delivered to the role recipients or accessed by the role occupants.

3. The Distributed RSA Scheme

This section presents the proposed key management scheme which splits a private key into two shares which are distributed to the role occupant and a role gatekeeper. This allows the two entities to hold the shares and to cooperate to digitally sign and decrypt messages. The role gatekeeper's share is then split again, one part is kept secret to the gatekeeper and the other part is given to an auditor. The auditor's key may be made public or kept secret, depending upon the wishes of the organisation for who can perform an audit of the signatures.

Let $< e_r, N_r >$ and $< d_r, N_r >$ be the role pubic key and the role private key respectively of role r, where modulus $N_r = p \times q$ and $\phi(N_r) = (p-1) \times (q-1)$, p and q are primes.

Let the occupant j of role r be denoted as $o(r,j)$. The occupant's key share is $k_{r,j}$, and the Role Gate Keeper's key share is $s_{r,j}$ and $v_{r,j}$. The share $v_{r,j}$ is the Auditor's verification key share. The key shares are generated as follows.

$$\begin{aligned} s_{r,j} &= f(r,j,\mathcal{S}_r) \\ v_{r,j} &= f(r,j,\mathcal{V}_r) \\ k_{r,j} &= d_r - s_{r,j} \times v_{r,j} \pmod{\phi(N_r)} \end{aligned}$$

where $f(x,y,z)$ is a public function that is easy to compute but is difficult to reverse. S_r is a secret managed by the Role Keeper for role r . V_r is a random number known to the Auditors and Role Keeper for role r . The role occupant can construct a private key $< k_{r,j}, N_r >$, the Gate Keeper can construct corresponding private keys $< s_{r,j}, N_r >$ and $< v_{r,j}, N_r >$. The Auditors can generate the verification key $< v_{r,j}, N_r >$ for every $o(r,j)$.

3.1. Signature Generation

The signing is a three phase process, including role occupant (RO) Signing, Role Keeper (RK) Signing, and Combining Signatures.

RO Signing. The RO $o(r,j)$ produces an initial message signature $sg_{ro}(r,j,m) = m^{k_{r,j}} \pmod{N_r}$. In other words, the role occupant $o(r,j)$ use his own key $< k_{r,j}, N_r >$ to sign the message.

Keeper signing. The RO computes a signature $sg_{kp}(r,j,m) = m^{s_{r,j}} \pmod{N_r}$ by using the key $< s_{r,j}, N_r >$.

Combining signatures. The signatures produced above can be combined together as the final signature $sg(r,m)$. $sg(r,m) = sg_{ro}(r,j,m) \times sg_{kp}(r,j,m)^{v_{r,j}} \pmod{N_r} = m^{d_r} \pmod{N_r}$. The newly produced signature $sg(r,m)$ is the same as the normal signature generated by using the private key $< d_r, N_r >$.

3.2. Decryption

A message m is encrypted for the role r using r's public key $< e_r, N_r >$ as follows. $em = m^{e_r} \pmod{N_r}$. Decrypting em involves the RK Decryption, the RO Decryption and Reconstructing the plaintext.

Keeper Decryption. The RK partially decrypts the message in two steps. First, the RK uses the key $< s_{r,j}, N_r >$ to perform a normal RSA decryption as below. $\widehat{em}_{kp(r,j)} = em^{s_{r,j}} \pmod{N_r}$. Then the RK performs a second RSA decryption using the verification key $< v_{r,j}, N_r >$. $em_{kp(r,j)} = (\widehat{em}_{kp(r,j)})^{v_{r,j}} \pmod{N_r}$.

RO Decryption. The RO decrypts the original message using the key $< k_{r,j}, N_r >$ using normal RSA decryption. $em_{o(r,j)} = em^{k_{r,j}} \pmod{N_r}$.

Plaintext Construction. With the RK Decryption result $em_{kp(r,j)}$, and the RO decryption result $em_{o(r,j)}$, the plaintext $\overline{m}$ can be reconstructed. $\overline{m} = em_{kp(r,j)} \times em_{o(r,j)} \pmod{N_r} = m \pmod{N_r}$.

In summary, decryption needs the role occupant and the keeper to each decrypt the message into two partially decrypted messages. The two partially decrypted messages are then used to reconstruct the plaintext.

3.3. Audit

The auditing process verifies whether the RO signature is generated by the claimed RO based on the specified message. In order to perform an audit, the Auditor needs to know the original message, the ID of the role occupant, the ID of the role, the partial signatures generated by the role occupant and the Role Gatekeeper, and V_r.

Assuming that the claimed identity of the role occupant is fake, let m denote the message hash that is signed, $sg_{kp}(r, j^{'}, m)$ is the signature generated by the keeper using identity $o(r, j^{'})$, $sg_{ro}(r, j^{''}, m)$ is the signature generated by the role occupant using the identity $o(r, j^{''})$, and $o(r, j)$ is claimed to be the identity of the original signer. The audit service will construct a role signature $m_v(r, j, m) = sg_{ro}(r, j^{''}, m) \times sg_{kp}(r, j^{'}, m)^{v_{r,j}} \pmod{N_r} = m^{(k_{{r,j}^{''}} + s_{{r,j}^{'}} \times v_{r,j})} \pmod{N_r}$.

Then the audit service can compute a hash value h_v by decrypting $m_v(r, j, m)$. $h_v = m_v(r, j, m)^{e_r} \pmod{N_r} = m^{(k_{{r,j}^{''}} + s_{{r,j}^{'}} \times v_{r,j}) \times e_r} \pmod{N_r}$.

- If and only if $j = j^{'} = j^{''}$ then $h_v = m^{(k_{r,j} + s_{r,j} \times v_{r,j}) \times e_r} \pmod{N}_r = m$, which means that if the given signatures of the keeper and the role occupant are produced based on the same identity as recorded by the audit trail, then the verification process succeeds with $h_v = m$. Thus it is believed that the identity of the role occupant is $o(r, j)$.
- The other case is, if $j \neq j^{'}$ or $j \neq j^{''}$ then $h_v = m^{(k_{{r,j}^{''}} + s_{{r,j}^{'}} \times v_{r,j}) \times e_r} \pmod{N}_r \neq m$. This means that at least one of the given signatures is not produced based on $o(r, j)$, and a fraud took place. Similar conclusions can be made over role identities.

In summary, the role occupant identity is believed to be true only when the audit process can use the claimed identity of the role occupant to construct a valid role signature based on the information provided by the audit log.

4. Managed Messaging System Architecture

The system architecture presented in Figure 1 constructs a SRBM system based on the key management scheme described in Section 3. The SRBM system architecture comprises of several main elements, including the Key Generation CA (KGCA), the Auditor, the Role Keeper (RK), and the role occupants (ROs). The KGCA is a trusted authority responsible for issuing keys, key shares and Certificates. The Auditor is an entity providing a verification service of all signatures produced by role occupants using their shares of role private keys. The RK participates in all role signature generation processes and all role message decryption processes. ROs are members of roles that perform messaging actions on behalf of roles.

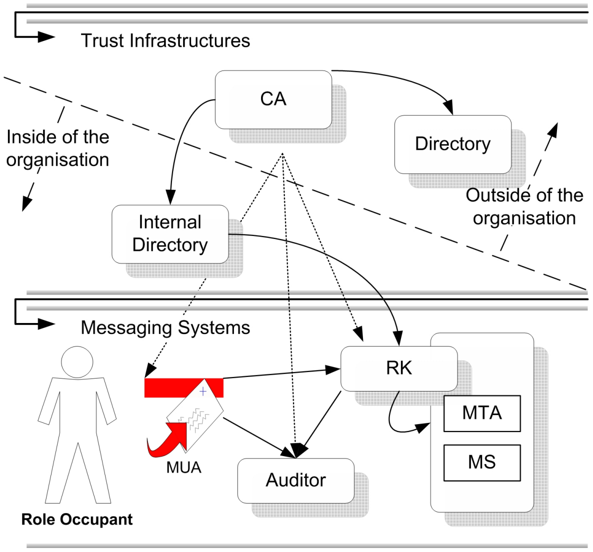

Figure 1: SRBM Architecture

4.1. Main components

The KGCA generates keys based on role private keys as described in Section 3 for roles, ROs and RKs. The KGCA generates thress types of keys or secrets, which are RO key shares, Auditor secrets, and RK secrets.

ROs send and read messages through Message User Agents (MUAs). All messages will be delivered to the role Keeper before they are sent through the MTA or before they are accessed from the MS. The Message Transfer Agent (MTA) is responsible for delivering messages to either a local Message Store (MS) or a remote MTA via the SMTP protocol [8]. The MS is a local database that stores all messages and accepts either the POP3 protocol [11] or the IMAP4 [6] protocol for message management and retrieval. Our implementation of managed messaging system will be based on these standards.

The RK is responsible for cooperating with the ROs to sign messages or decrypt messages. On signing messages, ROs will submit their initial signatures with the original messages. The RK will then re-sign the messages and combine the new signatures with the ROs' signatures, which will then produce valid role signatures. The process is described in Section 3.1. On decrypting messages, the RK will first conduct the keeper decryption and provide the result to the RO, then the RO can perform the RO Decryption using his key share, and reconstruct the plaintext. The decryption process has been elaborated in Section 3.2.

The Auditor is responsible for verifying the identity of role occupants who initiated the process of message signing when necessary. The Auditor has access to the role verification secret. A log is assumed to be produced by the system which contains the original messages (or the hashes of the messages), RO signatures, the RK signatures, and the RO identities. Based on all the above information, the Auditor can verify all RO' signatures using the role verification secrets. The verification process is described in Section 3.3.

4.2. Keys and Secrets

Three types of keys and secrets are generated by the KGCA, including the RO key shares, the role verification secret, and the RK secrets.

RO key shares are delivered to every corresponding individual RO. ROs may have multiple RO key shares if they hold multiple role memberships, and they are responsible for safely managing their RO key shares. Unauthorised disclosure of a RO's key shares will enable a malicious user to impersonate that RO.

The Auditor secrets are used by the system to verify RO identities recorded in the log. The role verification secret can be made either public within the organisation, which allows every one in the organisation to perform verification, or available to the RK and the Auditor only.

RK secrets are used by the RK to help ROs to sign or decrypt messages. The RK secrets must be private to the RK. Revealing the RK secrets to any RO will enable the RO to reconstruct the role private key and sign or decrypt messages without the aid of the RK.

4.3. Messaging Process

Secure messages are represented in S/MIME format [12] to provide protection for the information. S/MIME allows messages to be signed and encrypted in any sequence.

To send a message on behalf of a role using the proposed scheme, a RO needs to compose a S/MIME message and sign the message using her role key share. The message will be sent to the RK via the MUA. On receiving an outgoing message from a RO, the RK will reprocess the message, and convert the RO's signature into a valid role signature and replace the RO's signature in the message. The process is specified in detail in Section 3.2.

To access a role message which is encrypted to the role, a RO will need to submit a request to RK. RK will then decrypt the message based on the secret it holds and the RO's identity. The decryption result will be returned to the RO with the original encrypted message. Then the RO can decrypt the message using her role key share and construct the plaintext. The process is specified in more detail in Section 3.1.

When communication is required to be encrypted between the RO and the RK, all information can be encrypted to the role, and then be partially decrypted before being transferred to the other end. The recipient can reconstruct the information in the way explained in Section 3.2.

Outgoing messages can not be triple wrapped (e.g. signed then encrypted then signed with both signatures are role signatues) unless extra protocol interactions are provided, because it is not possible to generate a valid outer signature or partial outer signature without completing the inner signature first. We are dubious if the benefits of triple wrapping would outweigh the costs of such a complex interaction.

4.4. Audit and Security Control

The audit service is performed by the Auditor. To enable the Auditor to audit a role occupant's identity, the system must produce a log of all signing actions. For every signing, the log must contain the digest algorithm, the signing algorithm, the message digest, the RO's partial signature, RK's partial signature, and the claimed identity of the RO. The Auditor will follow the algorithm presented in 3.3 to verify if the claimed identity is authentic.

The SRBM system allows role occupants to send or access role messages, but all messaging must be conducted with the cooperation of the Role Gate Keeper. Any message processing that is performed by a role occupant which bypasses the Role Gate Keeper will not be able to be authenticated as originating from that role. Thus all authorised role based messages will need to pass through the Role Gate Keeper. A Role Gate Keeper can be further endowed with security control mechanisms so as to enforce local security policies on the messages. Local security controls may constrain the dissemination of internal documents, limit the scope of legitimate recipients, filter or remove unauthorised information in messages, or even prevent role occupants from accessing messages if the incoming messages do not meet the necessary security criteria.

4.5. Revocation

Role membership assignments need to be dynamic. Revocation of membership means that ROs can be deprived of their role memberships. Membership revocation will stop the RO from performing actions on behalf of the Role, thus the revoked RO will not be able to sign and send messages or decrypt and access messages on behalf of the role.

Role membership is usually revoked by a corresponding management authority. Role membership revocation needs only to be distributed to the related RK. Once informed, the

RK will refuse to cooperate with the RO whose membership has been revoked. Since a RO has to gain the cooperation of the related RK in order to send or access role messages, non-cooperation of the RK stops the revoked role occupant from performing the corresponding role. Note that the revocation need not remove the RO's key share from the RO, since it may still be used by the RO, but to no good effect. Furthermore the role public key and role private key can remain unchanged, since others may still use the role public key to validate role signatures and encrypt for the role, whilst the private key shares held by all the other role occupants are still valid.

5. Security Analysis

Potential attacks to the SRBM systems includes compromising role private keys, spoofing role occupant identities, and reusing revoked role occupant identities.

The ability to recompute the private key of the Role leads to the ability to generate messages originating from the role without the RK's involvement, hence no gatekeeping authorisation and security control would be performed. The ability to recompute the private key share of a role occupant would lead to the ability of an attacker to masquerade as the role occupant.

In this scheme the Role's private key is shared between the RO and the RK in such a way that each share of the private key is effectively a random number, assuming f is a hash function randomly mapping inputs into an integer. Thus it is not possible to use interpolation techniques to recompute the private key of the role or role occupants, even if any number of role occupants collude. Similarly, role occupants will not be able to perform any analysis of role private keys based on the key shares they hold for different roles.

A RK on its own cannot recompute the role private key, as it has only the RK secret shared with the CA. Only when a RO colludes with the RK can they recompute the role private key.

Compared with the security of keeping role private keys in complete (unshared) form, it is safer to have the role private keys split and stored in a distributed manner. Any attempt to reveal a role private key has to compromise both a RO and the RK in order to access both key shares.

It is shown in above that it is not trivial to compromise a RO key share. Assuming an attacker collude with the RK and tries to spoof $o(r,j)$ to be $o(r,i)$. RO $o(r,j)$ produces a signature $sg_{ro}(r,j,m)$ and submits it to the RK.

On the re-signing stage, the RK produces a spoof exponent $spoof_{r,j,i} = s_{r,j} \times v_{r,j} \times v_{r,i}^{-1}$, and then produce a keeper signature $sg^{'}_{kp}(r,j_i,m) = m^{spoof_{r,j,i}}$.

Thus $sg^{'}_{r,m}$ and $sg^{'}_{kp}(r,j_i,m)$ can succeed in passing the verification against the identity $o(r,i)$ as the Auditor can construct a valid role signature $sg^{'}_{r,m} = sg_{ro}(r,j,m) \times sg^{'}_{kp}(r,j_i,m)^{v_{r,i}} = m^{d_r} \pmod{N_r}$. This means spoofing $o(r,i)$ has succeeded. However, in order to achieve the spoof, it is necessary to compute $(v_{r,i})^{-1}$, but since the two primes used to construct N_r are known only to the PGCA, computing $(v_{r,i})^{-1}$ with them is as hard as the RSA problem. Thus assuming the RSA problem is hard, spoofing an RO is hard as well and is unlikely to succeed.

Conventional systems depend on publishing Certificate Revocation Lists (CRLs) to revoke role memberships. There is a time gap between the revocation of the membership the delivery of the CRLs to the relying party. This time gap allows revoked members to exercise their role without being noticed for a limited period of time.

With the proposed SRBM scheme, recipients judge role membership depending on role signatures only. Valid role signatures imply a legitimate role membership. To revoke role membership, the SRBM system only needs to notify the related RK about the revocation. The RK will then reject all cooperation requests from the revoked role member, preventing the RO from signing or decrypting role messages.

6. Related Work

Mont *et al* [10] and Chadwick *et al* [4] both proposed a model of role based messaging based on a trusted entity. These two models can provide effective role membership revocation, but the trusted entity is the central point of attack. Breaking into the trusted entity will compromise the security of the whole system.

Group signatures [5, 3] enable group members of a group to sign messages on behalf of the group anonymously. The signature can be verified using the group public key. Only the trusted authority with the group secret can identify the signer identity from a valid group signature. Group signatures can facilitate the signing process of role based messaging, but they fail to impose mandatory controls on the generation of signatures, as well as they fail to support data encryption and decryption.

Threshold cryptography [7] shares secrets among shareholders, and requires at least any k out of t shareholders to cooperate together to reconstruct the primary secret which is the original private key. Threshold cryptography promotes Ad Hoc cooperation between shareholders, which makes it difficult to impose mandatory security control.

Boneh et al. [1] presented an approach for fast certificate revocation. The approach split the users' RSA private keys into two shares and managed by the users and a semi-trusted entity respectively. To produce a valid signature or decrypt a message, users have to acquire cooperation from the semi-trust entity. Boneh et al. implemented the semi-trusted entity to check the validity of users' keys to achieve fast certificate revocation. The difference between Boneh's system and ours is that, our system is designed to share role identities and support security control, while Boneh's system is designed for protecting individuals' private keys. Our system is superior to Boneh et al. because it can be built using a single secret for all role occupants, and it allows an Auditor to check which occupants acted in a given role.

A distributed algorithm to produce a RSA key pair and its key shares jointly by a number of peers has been proposed [9, 2]. Peers participating in the generation do no know the private key or the key shares that others hold. This system is superior to ours in that it does not require a CA to know the private keys of the role occupants.

7. Conclusions

This paper has developed a distributed RSA scheme for secure role based messaging, and presented the system architecture of the proposed scheme. A security analysis is also conducted to show the strength of the proposed scheme.

The advantages of the proposed scheme are five fold. Firstly, the scheme promotes distributed security. Keys are shared and distributed. Distributed security increases the overall system security and protects the systems against attacks by outsiders. Secondly, the scheme enables the auditor to verify the actual identity of any user acting in a given role, by checking the audit log of partial signatures. Thirdly, the scheme supports fast membership revocation. Only the Role Keeper needs to be informed when a role occupant is removed from a role, and from then onwards the RO will not be able to effectively use their role key share. Fourthly, the scheme supports the imposition of mandatory security controls. The Role Keeper is a critical entity involved in all communications on behalf of roles, therefore it can be employed as a checking point for imposing mandatory security controls. Fifthly, the scheme reduces the complexity of key share management schemes by using keeper secrets and verification secrets from which key shares can be automatically generated.

The proposed scheme and model is not without its limitations. Due to the way that role signatures are generated, requiring the cooperation of both the role occupants and the Role Keeper, roles can not send out a triple wrapped S/MIME messages without new more complex protocol interactions (see Section 4.3, unless the signatures are produced by different identities. As the proposed scheme is based on RSA cryptographic algorithms, it is theoretically as hard to break as RSA cryptography, and is also theoretically as efficient as RSA cryptography. On the other hand, any valid attacks on RSA cryptography will be valid attacks on the proposed scheme.

The contribution of this work is as follows. Firstly, we identify the requirements of secure role based messaging by recognising the need for distributed security, fast membership revocation, mandatory security controls and identification of role participation. Secondly, we propose a distributed RSA scheme which caters for these requirements. The scheme is able to utilise system logs to identity role participation, which is not provided by Boneh et al [1]. Thirdly, we demonstrate the application of the proposed scheme by developing a SRBM system architecture and depicting the communication process. Fourthly, we provide a security analysis of the proposed scheme showing the security performance of the proposed scheme.

Future work will focus on further reducing the key management complexity of the proposed scheme, and will also investigate the possibility of reducing the number of keys held by a single RO which is expected to improve the user friendliness of the scheme and simplify the user's task of key management. Building the scheme based on alternative cryptography systems other than RSA will also be studied.

References

[1] D. Boneh, X. Ding, G. Tsudik, and C. Wong. A Method for Fast Revocation of Public Key Certificates and Security Capabilities. In *Proceedings of the 10th USENIX Security Symposium*, 2001.

[2] D. Boneh and M. Franklin. Efficient Generation of Shared RSA Keys. *J. ACM*, 48(4):702–722, 2001.

[3] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *the 17th Annual International Cryptology Conference on Advances in Cryptology*, London, UK, 1997.

[4] D. Chadwick, G. Lunt, and G. Zhao. Secure Role based Messaging. In *the Eighth IFIP Conference on Communications and Multimedia Security*, Windermere, UK, September 2004.

[5] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, pages 257–265, 1991.

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

[7] P. Gemmell. An Introduction to Threshold Cryptography. *CryptoBytes*, 2(3), Winter 1997.

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

[9] M. Malkin, T. Wu, , and D. Boneh. Experimenting with Shared Generation of RSA keys. In *SNDSS 1999*, San Diego, California, February 1999.

[10] 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.

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

[12] B. Ramsdell. RFC 3851 - Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification. Request For Comment, Network Working Group, July 2004.