Privilege Messaging: An Authorization Framework over Email Infrastructure

Brent ByungHoon Kang, Gautam Singaraju, and Sumeet Jain
– University of North Carolina at Charlotte

ABSTRACT

The current email infrastructure is burdened by multiple resource constraints and a plethora of security issues. Apart from the fact that email users are spending more time and effort sifting through unsolicited emails, more serious problems such as Phishing are on the rise. This can be attributed to a fundamental shortcoming in the current email infrastructure: a lack of an authorization framework. This allows any user to create content in anyone's mailbox. In this paper, we revisit the fundamental problem of non-existent authorization and discuss the design of an effective authorization service overlaying the existing email infrastructure. We propose Privilege Messaging (P-Messaging), a fine-granular authorization framework that operates on the principle that a sender requires a set of privileges in order to send messages, simultaneously enables the receiver's infrastructure server to verify the messages before accepting it. We present a prototype implementation and discuss its benefits. An automatic classification of email can be effectively performed based on the privilege-tag. Privilege-tag can provide flexible and fine-granular reputation management than current domain-based solutions. The use of privilege-tag as entry ID in a white-list can be more manageable than the use of individual email address. Finally, the privilege-tag can be used as an email header, retaining the benefits of currently deployed MTA architecture, namely reliability and flexibility.

Introduction

Email, a simple and cost effective messaging technology, has become the universal mode of interaction over the Internet. Undeniably, email is so established that day to day commercial and non-commercial activities are contingent upon its reliable operation. However, the Internet explosion has also introduced a variety of new risks and threats. Unsolicited email has reached epidemic proportions, severely limiting the usability of the current email infrastructure with non-essential resource consumption.

Spam, interchangeably referred to as Unsolicited Bulk Email and/or Unsolicited Commercial Email, is just one of the key threats faced by the current email infrastructure. Conveying worthless or fraudulent information, spam adversely affects the recipient in terms of time and money by encumbering limited resources such as server space [7]. Apart from the end user, each of the numerous relay stations also pays toward the resources for routing, bandwidth and memory.

Another threat, Phishing [13], fools recipients into divulging their financial information by redirecting them to a masqueraded site. The US Electronic Payments Association estimated that world wide financial losses due to Phishing reached approximately $500 million in 2004 [14].

In addition to the Spam and Phishing problem existent in the current email infrastructure, users spend significant amounts of their time and money classifying and labeling their emails. There has been only limited automatic classification mechanisms based on user-specified rules. In other words, there does not exist a Quality of Service (QoS) mechanism for users to classify emails based upon the email's relevance and importance. The classification and the relevance of email should be based on associated trust information, rather than a classification based on its content and properties.

Recent legislations, such as CAN-SPAM [4], introduced to curtail unsolicited emails, have been unsuccessful due to the impracticality of monitoring large numbers of email communications all over the world. In a desperate effort to curtail spam, some organizations have undertaken measures such as maintaining an isolated internal email infrastructure. Other prevalent techniques include the use of spam filters or the blacklisting of email accounts. These systems provide an excellent mechanism to weed out most unsolicited emails; however, they might, potentially, blacklist a legitimate email account, rendering it ineffective. Thus, the financial losses to organizations are not only due to the unsolicited emails but also from legitimate emails being falsely classified as unsolicited email [6].

The fundamental reason for the threats faced by the current email infrastructure comes from the absence of an authorization framework. Currently, any user on the Internet can send messages to another user's mailbox with no mechanism that differentiates between authorized and unauthorized senders. This is the primary reason for the proliferation of both illegal and uninvited content. There is an obvious need for a system that allows only authorized emails to be accepted by the receiver.

We propose a Privilege Messaging (P-Messaging) Framework, which enables a legitimate sender to send email and the receiver's server to verify the email's privileges before the content is created in the receiver's mailbox. A privilege can be viewed as a credential associated with a group of users. An email can hold multiple privileges, for instance, one as a faculty member and another as a USENIX member. Furthermore, P-Messaging provides trust-based messaging, given that under the P-Messaging framework each email is signed (i.e., vouched as to its validity) by the sender's P-Messaging infrastructure server. The trust-based mechanism supports the authorization mechanism through the creation and maintenance of a Circle of Trust among the P-Messaging providers.

The rest of the paper is organized as follows: the next section examines the related work in this area. Subsequent sections detail the design and architecture of P-Messaging, describe the benefits of P-Messaging, and discuss usage scenarios. Next is the evaluation of P-Messaging, future work, and finally the conclusion.

Related Work

Network-based email communication has existed since the early days of ARPANET [9] enabling a small, close-knit group to communicate electronically. Today, even with the extensive usage of email infrastructure [7], there exists no authorization service; hence, there is no way to guard against fraudulent mailers sending unsolicited messages to another user. To combat this, multiple filtering technologies have been developed that weed out most, but not all of the unsolicited email.

Figure 1 shows the comparison between the domain-based solutions and the user based white-list maintenance. Domain based solutions have credential information of the mail servers in their DNS entries; the credentials that are available are varied and dependent upon the solution adopted. The advantage of Domain based solution is that these systems allow trust based communication among the sender and the receiver. However, these publicly available credentials are not validated by a common trusted third party. Spammers and Phishers have setup their own domains and have continued to spam. Domain based solutions provide little QoS: we define QoS as automatic classification of the emails based on both the sender credentials associated with the email and the credentials that are accepted by the receiver.

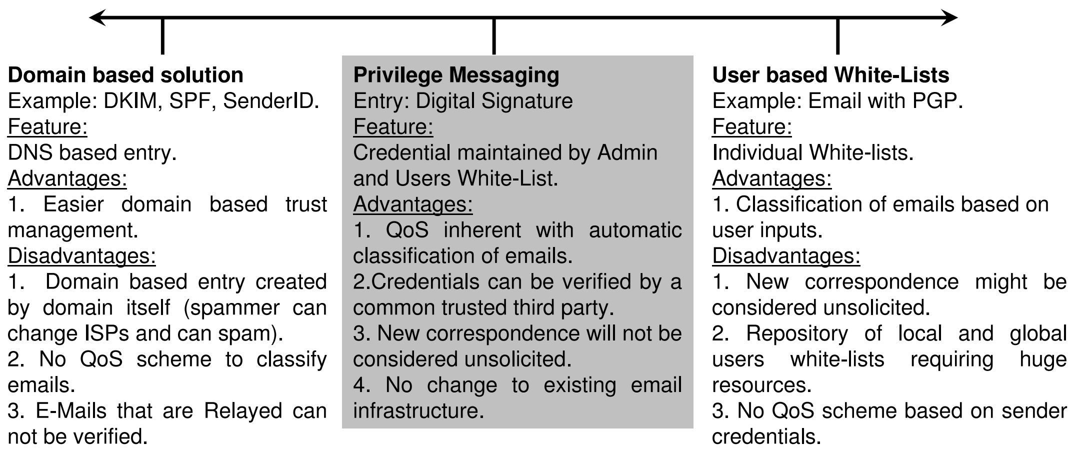
In case of the white-list based solutions, the email classification is performed by checking the sender's email IDs against the white-list maintained by the receiver. A new correspondent might be considered an unsolicited sender unless the email ID is enlisted into the white-list beforehand. The white-list size can grow unbounded because each user needs to list all the email IDs that they deem valid. Moreover, if the email ID needs to be revoked or changed, it requires all the user's contacts to update their white-lists.

These systems can be classified based on a sender's information and/or analyzing the email's contents. We categorize them into two categories that scrutinize the message based on a) the characteristics of the email content and blacklisted email IDs; and b) the sender domain's credential.

Classification Based on Email's Content and Collaborative Blacklisting

Word filters [1] search for patterns and remove the most obvious spam; however, spammers have often circumvented word filters by using misspelled words. Thus, word filters requires regular updates with commonly misspelled words in unsolicited emails. Rule-Based scoring mechanisms check for keywords, but use rules to analyze emails. For instance, depending on the score received by a particular email, it is classified as either spam or not spam. Bayesian Filters, on the other hand, perform lexicographical and statistical analysis on the email for words and/or phrases depending on the recipient's previous emails.

Incorporating user feedback at the MTA level forms the basis of another technique: collaborative filters [8]. With collaborative filters, an unsolicited email is filtered at the MTA with users' feedback about



Domain based solution
Example: DKIM, SPF, SenderID.
Feature:
DNS based entry.
Advantages:
1. Easier domain based trust management.
Disadvantages:
1. Domain based entry created by domain itself (spammer can change ISPs and can spam).
2. No QoS scheme to classify emails.
3. E-Mails that are Relayed can not be verified.

Privilege Messaging
Entry: Digital Signature
Feature:
Credential maintained by Admin and Users White-List.
Advantages:
1. QoS inherent with automatic classification of emails.
2.Credentials can be verified by a common trusted third party.
3. New correspondence will not be considered unsolicited.
4. No change to existing email infrastructure.

User based White-Lists
Example: Email with PGP.
Feature:
Individual White-lists.
Advantages:
1. Classification of emails based on user inputs.
Disadvantages:
1. New correspondence might be considered unsolicited.
2. Repository of local and global users white-lists requiring huge resources.
3. No QoS scheme based on sender credentials.

Figure 1: Comparison of Privilege Messaging with current technologies.

falsely classified emails, hence developing a smarter spam filter. However, an email that is marked as spam by a user might not be considered spam by another user.

A combination of different techniques provides a reliable means to classify an email as spam [11]. SpamGuru [18] employs multiple techniques such as word filters and Chung-Kwei. Chung-Kwei uses a pattern discovery technique, to classify unsolicited emails.

Other techniques exist such as HoneySpam that borrows the idea of honeypots to email infrastructure, the Social Network based classification, and a new email delivery framework that proposes receiver-pull instead of sender-push [3, 5, 15]. However, these systems do not address the essential problem of unrestricted access to others' mailboxes.

Classification Based on Sender Credential

Blacklist IP is comparatively simpler and computationally less intensive than other techniques. This process keeps a list of IP addresses identified as spammers and another white-list for legitimate users. Realtime Blackhole List (RBL) [17] works similar to Blacklist IP, but RBLs are not manually updated by individual organizations. Instead, RBL operators maintain public RBLs to which organizations subscribe.

PGP [16, 22] and SenderID [12] techniques allow verification of a sender's email address based on the sender's domain credentials. PGP is a fine-granular service containing a list of individual users' public keys. User contact management based on PGP keys can provide the benefits of identifying trusted peer correspondents as well as verifying the email integrity. However, utilizing PGP-based authentication techniques incurs the overhead of requiring individual users to maintain such lists. A new correspondence might be considered unsolicited, unless the email ID is enlisted beforehand. Also, the size of the white-list can grow unbounded because the local and global white-lists may need to list all the legitimate email IDs on the Internet.

SenderID addresses the problem of spamming and Phishing by validating an email's origin, i.e., by verifying the IP address presented by the email against the sending domain. This validation is performed by comparing the sender's IP address against the registered domain's mail servers. SenderID is a coarse-granular service validating a sender's domain. By coarse-granular, we mean that there is only one credential available for the entire domain.

Sender Policy Framework (SPF) [20] is a technique that has been introduced to prevent email forgery. In SPF, the mail servers are identified by DNS entries which receivers use to validate the sender's MTA. The DNS records also indicate the sender's adopted policy, for instance, the list of mail servers allowed to send email from a domain. At the receiver's end, when the mail is received, the receiver checks the sender's policy specified through their DNS records. For example, if the sender's email server is not the one specified in the policy, the email is considered unsolicited.

Domain Key Identified Mail (DKIM) [2] attempts to reduce the traffic on the network by publishing the mail server's public keys through DNS records. Each domain creates and publishes its public key. Each email sent henceforth is signed by the sender's mail server. The receiver's mail server can verify the digital signature by retrieving the public key of the sender from their DNS records. Thus, the sender's domain information, as presented by the email, is validated with the actual domain. However, in DKIM, the public-private key pair is generated by the domain itself. Additional security can be provided by publishing the keys at a Certificate Authority (CA) [21]. However, presently DKIM does not enforce this restriction.

Although there are major advantages with these systems that classify emails based on sender credentials, there exist distinct disadvantages. Firstly, a significant portion of spam is usually sent from perfectly legitimate domains. For instance, the spammer can spam by creating a new domain with the required credentials, allowing them to send spam. Secondly, relaying emails cannot be performed because the sender's domain information in the email will be different from that of the relaying domain.

These systems motivate a need for a fine-granular service: finer than domain-based solutions and coarser than white-list management. A coarse-granular service removes the need to maintain a local white-list, whereas a fine-granular service allows each email to be associated with multiple privileges rather than just one as in the domain-based solution. Using a fine-granular service also allows a negative reputation to be restricted within a small group of users rather than scaling to the complete domain. Hence, there is a need for a framework that allows users to explicitly specify which group of email IDs has the appropriate credentials to create content in their own mailbox. In other words, an authorization framework is required that verifies the sender and based on the authorization presented, be able to classify mails for them. In order to provide such a scalable authorization framework, we introduce Privilege Messaging.

Privilege Messaging

The systems described in the previous section are some of the techniques that the current email infrastructure has adopted in order to classify emails as spam and legitimate emails. These systems identify spam based on either the email contents or the sender domain. However, they do not provide authorization services before the content is created in receiver's mailbox. Moreover, a domain is responsible for the creation and maintenance of its own authentication credentials. Though such a technique considerably restricts a spammer by requiring extensive computation before sending a bulk email, such a mechanism does not allow for qualified trust to be placed by the receiver.

Revisiting the fundamental issue of non-existent authorization problem is a first step toward designing a system that can be an effective solution to the security loopholes in the email infrastructure. Furthermore, any attempt to provide a better email infrastructure should address this non-existent authorization. To combat proliferation of unwanted traffic using an authorization service, a new system must be designed with a scalable architecture to enable fine-granular control for sending and verification of emails.

Thus, the need for P-Messaging, a system that provides authorization, cannot be emphasized more. P-Messaging stipulates that the verification of the privileges held by the email will determine its authenticity and relevance to a particular user. Each email ID can be assigned multiple privileges. For instance, each email ID would be associated with various groups, such as one for each department in a school or one for each project team in a development environment. The granularity of a group can be defined by the users. Using P-Messaging, an email ID, based on the privileges held, can send and receive email with authorized users with the help of a privilege. As each privilege has a PKI key pair, digital signatures are used to verify the privileges associated with the emails.

Capable of supporting different Message Transfer Agents (MTA), P-Messaging provides both better security and improved QoS over the present email infrastructure. The MTAs provide the essential services of relaying, spooling, queuing and sending emails; P-Messaging is a gradual process of introducing the authorization feature. To support such deployment, P-Messaging framework along with the MTAs provides trust based communications.

In most cases, unsolicited email arises from mail servers that are either (i) Zombie or (ii) Unauthorized servers set up temporarily. Towards controlling the unwanted traffic from them, creation and maintenance of a Circle of Trust (CoT) among P-Messaging providers is essential. A CoT is a trust relationship formed between a sender and a receiver due to the trust placed on them by a common third party entity, such as a Certificate Authority (CA). Using CoT, qualified trust can be placed by a receiver on a sender since the CA is responsible for maintaining the trust among the servers. Each of these servers in turn maintains a level of trust on each privilege they maintain.

With the help of the CoT, any email received that is not trusted is placed in a no-privilege class, which forms the lowest available privilege class. P-Messaging classifies emails for the receivers into multiple classes; only emails from the members of CoT could be classified. Any attempt to create unsolicited content will result in the trust being revoked. Members of the CoT are capable of informing the admin of a particular system about possible spam arising from it. Hence, to be able to use Privilege Messaging, each server in the CoT should continually strive to remain in the CoT. Thus, the CoT creates trust among P-Messaging Providers. The trust maintenance among the members will be discussed later.

The following sections discuss the functional components of P-Messaging, architecture for CoT, followed by the architectures of P-Messaging and finally management of the P-Messaging providers.

P-Messaging: Functional Components

This subsection discusses the functional components of P-Messaging. These components consist of the:

1. P-Messaging Server
2. P-Messaging Privilege Verifier
3. P-Messaging Manager
4. P-Messaging Trust Authority

P-Messaging Server

Architecturally, P-Messaging Server (P-Server) is a sandbox before the MTA. To send an email, the users interact with the P-Server, which in turn interacts with the MTA. After receiving an email from a user, the P-Server validates the user and attaches the credentials (i.e., privileges) to the message. Finally, the message along with the credentials is transmitted through the MTA.

The P-Server provides different services: (i) User Authentication Service, (ii) Privilege Lookup Service, (iii) Message Signing Service and (iv) Privilege Admin Service. The User Authentication Service provides a mechanism to authenticate a user to the P-Server. This Authentication Service can be password-based authentication scheme. Privilege Lookup Service, based on a rule-based engine, allows senders to look up their privileges. Once the privilege is selected, the Message Signing Service signs the email based on the privilege and then with the P-Server's key. The Privilege Admin Service, on the other hand, provides an administrative interface to the privilege administrators to add and revoke users and privileges.

P-Messaging Privilege Verifier

P-Messaging Privilege Verifier (P-Verifier) provides Privilege Verification and email classification services. Upon receiving a privilege signed email, P-Verifier verifies and based on the authorization presented, classifies the email.

The P-Verifier provides two services: (i) Message Authorization Service and (ii) Privilege-list Maintenance. The Message Authorization Service verifies the validity of emails and based on the privileges accepted by a user, classifies the mails into different privilege classes. The Privilege-List Maintenance service provides an interface for user to maintain a list of privileges accepted.

P-Messaging Manager

The P-Messaging Manager is a component that connects to the Privilege Admin service of the P-Server and provides an interface to add and remove both users and privileges. As discussed above, each P-Server has an administrator who has the capability to maintain the privileges. The administrator has the ability to add and remove privileges. While creating a privilege, the administrator nominates a privilege-owner. The privilege-owner is responsible for maintenance of the users of the privilege. The privilege-owner has the capability to add and delete users from the privilege.

P-Messaging Trust Authority

The P-Messaging Trust Authority is the entity that creates a CoT among P-Servers by providing a certificate to each P-Server. With the help of a digital signature-based mechanism [10], the trust on the senders' P-Server can be verified by the receivers.

Circle of Trust in P-Messaging

P-Messaging provides the capability of verifying a P-Server by peer P-Servers before placing any trust on them with the help of CoT. The CoT among the P-Servers is formed with the help of PKI infrastructure. Honoring a P-Server's privileges, i.e., accepting privileged email to be valid, across domains is dependent upon the maintenance of the trust placed upon the P-Server by the P-Messaging Trust Authority. If a P-Server sends an unsolicited email, the trust placed on it by the Trust Authority can be revoked. To be able to send authorized emails to other P-Servers that are a part of the CoT, a P-Server would strive to be a part of the CoT.

With the help of privilege verification, a recipient can first authorize and then classify the incoming emails into the privilege classes. If an email whose privilege is not subscribed to by the user is received, it is placed into an underprivileged class. However, if a sender's P-Server is not in the CoT, the message is placed into the no-privilege class.

The following subsection describes the various methods in which a CoT is created and maintained in order to support Privilege Messaging.

Addition of a P-Server to the CoT

Figure 2 describes the hierarchical structure of P-Messaging, where the P-Messaging Trust Authority forms an entity that functions as a CA. We make the assumption that the P-Messaging Trust Authority is trusted by all. Each P-Server must receive a certificate from the P-Messaging Trust Authority that is used to verify the P-Server. Each P-Server in turn maintains or moderates multiple privileges. Each of the privileges has its own PKI key pair capable of signing the emails.

Creation of the CoT requires each P-Server to store the privilege's private keys in a secure manner: the key store that holds the private key of the privileges should be maintained securely; i.e., in case of an unauthorized access, the keys must be destroyed. In case of the unauthorized access of the private key of the privilege, the administrator can easily revoke the privilege and create a new PKI key pair for the privilege. We discuss revocation policy adopted a CoT in the next section.

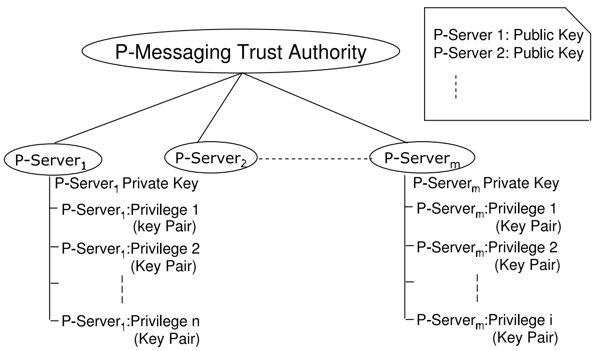

P-Messaging Trust Authority
P-Server 1: Public Key
P-Server 2: Public Key
P-Server1
P-Server2
P-Serverm
P-Server1 Private Key
P-Server1:Privilege 1 (key Pair)
P-Server1:Privilege 2 (Key Pair)
P-Server1:Privilege n (Key Pair)
P-Serverm Private Key
P-Serverm:Privilege 1 (Key Pair)
P-Serverm:Privilege 2 (Key Pair)
P-Serverm:Privilege i (Key Pair)

Figure 2: Circle of trust among the Privilege Servers. The P-Messaging Trust Authority allows a Privilege Server to be verified by other Privilege Servers.

Revocation of a P-Server from the CoT

The P-Messaging Trust Authority has the authority to revoke a P-Server based on an issuing agreement of the certificate. One of the attributes of the issuing agreement could be the number of instances an unsolicited mail is reported by users before revoking a P-Server. Once revoked, the P-Server can request a new certificate from the P-Messaging Trust Authority. The P-Messaging Trust Authority can place additional constraints on the P-Server before issuing a new certificate. This paper discusses the mechanism involved to revoke a P-Server; the pre-stipulated policies for revocation are beyond the scope of this paper.

Upon compromise of a privilege, the P-Server administrator is responsible for revoking that privilege. If the privilege is not revoked, the P-Server itself will be revoked from the CoT, thereby invalidating all the privileges held by it. Hence, it is contingent upon the P-Server administrators to maintain the trust placed upon it. A P-Server is responsible for maintaining the legitimate users for each privilege maintained by the P-Server and is delegated to the privilege-owner.

To reiterate, the negative characteristics of a privilege-user will flow upward to their privilege, where if the user is not revoked by the privilege, the privilege is revoked by the P-Server. If the P-Server would not revoke the privilege, the P-Messaging Trust Authority will revoke the P-Server.

Advantages of CoT

With the help of CoT, a P-Server can be verified by peer P-Servers. In other words, as shown in Figure 2, each P-Server acts as a CA for the multiple privileges maintained by it. With the help of CoT, each P-Server independently possesses the capability to create and maintain the privileges that are associated with it. Honoring of a privilege is based on the trust placed on the P-Server by the P-Messaging Trust Authority. This provides distributed authorization among P-Servers where each P-Server is capable of creating its own privileges. To reiterate, with the help of CoT, a scalable architecture with fine-granular email authorization can be provided.

P-Messaging Architecture

As discussed in earlier sections, P-Messaging provides message verification, thereby classifying emails based upon the privileges held. With the help of P-Messaging as shown in Figure 4, legitimate messages can be honored across domains where each domain is managing multiple P-Servers.

Privilege Messaging allows users to send and receive the messages. In the sender architecture, the P-Server attaches privileges on behalf of the user. In the receiver architecture, these privileges are verified before being delivered to the receiver. The following sections discuss the two architectures in further detail.

Sender Architecture

As shown in Figure 3, when sending an email the P-Server first verifies the sender, for instance, Bob. After verification of the user, the P-Server signs the mail with the privileges requested by Bob. The user, Bob, can select a privilege or the P-Server can select a privilege with the help of a simple rule-based engine. The privilege is selected from the Member List maintained for every user at the P-Server. Once the message is signed with the privilege, the message is then signed by the P-Server itself, before relaying it through the MTA to the users who accept the said privilege.

This way, a receiving P-Server can verify the sender P-Server and place trust on the P-Server and then verify the privilege. For example, when a P-Server is installed for a university, the P-Messaging Trust Authority creates a key pair for the P-Server that is securely transmitted. The university P-Server can then create multiple privileges, for example, faculty and student privileges.

For a receiver to accept a message, the receiver should honor the sender's privilege. Without this, the message that is sent cannot be classified into privilege classes but into the underprivileged class. These classes are described in later sections.

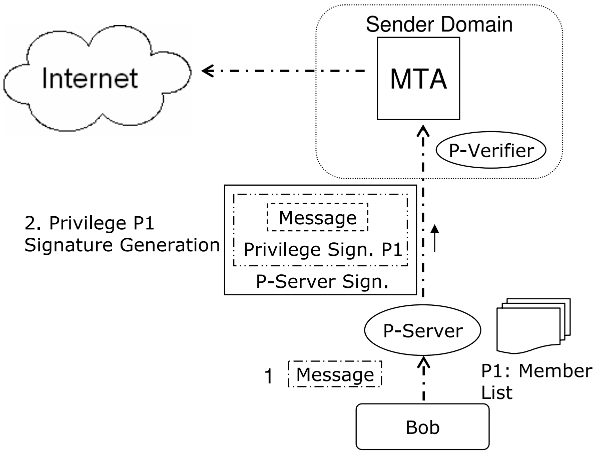


Figure 3: P-Messaging Sender Architecture: the sender Bob is verified, the P-Server signs the message using a privilege specified in the Member List. The mail is then sent from the P-Server to the MTA that relays the email.

Receiver Architecture

Figure 4 shows the receiver architecture in detail. On the receiver's domain, the MTA, upon receiving the email, verifies the privileges with the help of the P-Messaging Privilege Verifier. For verifying a mail, the P-Messaging Privilege Verifier first verifies the P-Server signature, thereby checking the authenticity of the P-Server in the CoT. Once the sender's P-Server is verified, the privilege's public key is retrieved from the P-Server and the email is verified.

Once the mail is verified, the next step is to place the mail into classes. This is performed by checking the user's Privilege List, which contains all the privileges that are accepted by a user. If the receiver honors a privilege, the mail is classified into the privilege class. The email classification is based on the privilege information in the email's header. If the mail is not

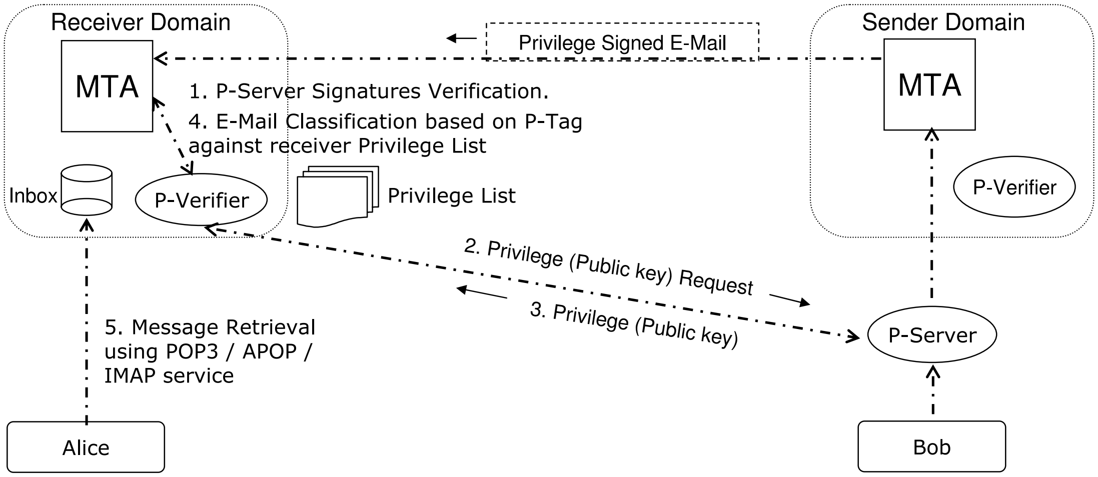

Receiver Domain
MTA
Privilege Signed E-Mail
Sender Domain
MTA
1. P-Server Signatures Verification.
4. E-Mail Classification based on P-Tag against receiver Privilege List
Inbox
P-Verifier
Privilege List
P-Verifier
2. Privilege (Public key) Request
3. Privilege (Public key)
5. Message Retrieval using POP3 / APOP / IMAP service
P-Server
Alice
Bob

Figure 4: P-Messaging Receiver Architecture: Once an email is received, the MTA passes the mail to the P-Verifier that looks up the public key of the privilege to verify the mail. Once the email is verified, it is classified according to the receiver's Privilege List.

honored by the user, the mail is classified into an underprivileged class. If, however, the message is not verified or signed, the message is placed into the unsigned class. We further discuss the privilege classes in the sections below.

To retrieve the message, as shown in Figure 4, the client, for instance Alice, connects to the mail server to retrieve the messages. Using the additional header information, any email client can display the information in any desired format. The mail clients can show the different 'inboxes,' where each inbox caters to a different class. In this way, the classification of the email into different classes provides users with the ability to view the messages according to the privileges accepted by them. This allows a faster lookup for the emails by classifying the emails at one location thereby providing QoS for the users.

Recursive Privilege

Another benefit of P-Messaging would be to request a privilege from another P-Server on the basis of a privilege that is held by the email. For example, as shown in Figure 5, a user in the UNCC domain requires a USENIX privilege with an email. The UNCC privilege would not be accepted by, suppose, LISA committee. However, on the basis of Bob's Privilege, the LISA privilege can be obtained. The LISA privilege can be used to communicate with other users of the Privilege -LISA. Figure 5 shows the Sender architecture for Recursive Privilege mechanism. The advantage of such a technique is that users can sign using their privileges across another administrative domain before sending an email. Another example would be a user using a free email ID to sign the mail using the class privilege to send the mail to the faculty who teaches the course at a university. A single mail can thus have multiple privileges, demonstrating to the receiver the sender's multiple credentials.

Recursive Privilege requires a user to be a member of a privilege across domains. The member list contains the list of members who are authorized to send an email using the privilege. A privilege can be created for each user for enabling cross domain privileges. This enables users to attach a privilege as a single user rather than the complete group. While requesting an email, the P-Server sends the Privilege-tag information of the first privilege. Instead of transmitting the complete message to the secondary P-Server, only the privilege information can be sent. This requires that only a small amount of information be transmitted over the wire to receive additional privileges. After verification of the privilege, the peer P-Server verifies the privileges and attaches its own privilege. Moreover, the verification of the privileges will be based on recursive verification of each privilege, allowing users to trace the order of privilege selection.

P-Messaging Classes

P-Messaging defines multiple Privilege Classes for the emails classified for the receivers. These classes take into consideration the credentials presented by the email as well as the receivers preferences. The user preferences indicate the list of the privileges that are further honored after the email has been verified. As described earlier, P-Messaging provides QoS with the help of this classification. We define the Privilege Classes into three categories.

Privileged Classes

Privileged classes contain the emails that are successfully verified and honored by the receiver, are placed. The emails can be further classified based on the privileges that it is associated with it.

Underprivileged Class

Underprivileged classes contain the emails that are verified, but the associated privileges are not honored

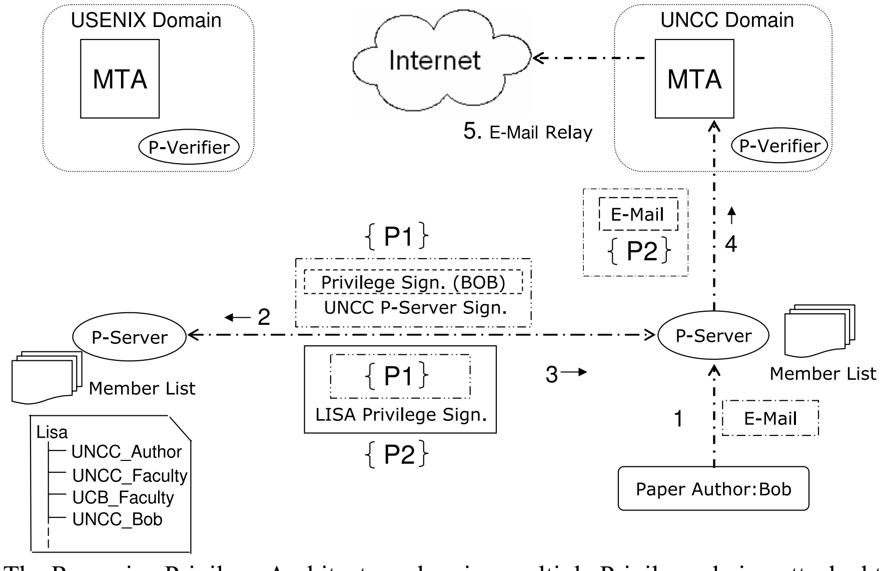


Figure 5: The Recursive Privilege Architecture showing multiple Privileges being attached to an email.

by the receiver. If a privilege presented by an email is deemed important, the receiver may subscribe to the privilege. An email from a sender with a privilege that is not honored will be placed in this class. Once a user honors a privilege, it will be placed into the Privileged Class.

No-privilege Class

The no-privilege classes form the lowest class among the privilege classes, where unsigned or emails whose authenticity cannot be ascertained are placed. As P-Messaging becomes widely accepted, the number of mails in the no-privilege class would be reduced.

Privilege Tag

Each email possesses credentials that allow the email to be classified into different classes. These credentials, referred to as Privilege Tag (P-Tag), provide the users with the information about the sender with the help of a digital signature. With the help of digital signature, Privilege Messaging demonstrates the authenticity of the email's origin.

In this section, we describe the format of the P-Tag and the various interfaces that are required to maintain the privilege.

Extensible P-Tag

As described above, the P-Tag information contains the privilege's digital signature. P-Messaging Tag management is extensible, so that each P-Server creates its own privileges. Each P-Server maintains privileges' public keys for other P-Servers to verify a privilege. Thus, each P-Server acts as a CA for the privileges it holds. The P-Tag format is as follows:

```
[P-Server]:[Privilege]
```

The P-Tag information is appended as a part of the email header. Conceptually, the following is the structure of a Privileged message:

```
{[Tagged Email] Privilege Signature}:
{[Privilege Signature]
                    P-Server signature}
```

The privilege signature is created on the email. The P-Server signs the Privilege Signature. Hence, to verify a privileged email, first the P-Server signature is verified. Once the P-Server signature is verified, the Privilege Signature needs to be verified. In the case of Recursive Privilege, the P-Tag information is shown below:

```
{[Tagged Email] Privilege Signature}:
{P-Tag 1}: ({P-Tag 1 }:P-Tag 2) ...
Where P-Tag n is {[Privilege Signature]
                   P-Server signature}
```

As discussed in sections above, in the recursive Privilege-tag assignment, multiple privileges are attached, based on the privileges already presented by it. The complete message need not be transmitted to the second P-Server, as only the P-tag information is needed to verify the sender of the privilege to identify and attach a new privilege to it. The next few sections describe the method for creating and maintaining the P-Tag information.

P-Tag Creation and Maintenance

As part of Privilege Management, apart from creation and maintenance of the privileges, a privilege-owner performs the tasks of adding and deleting/revocation of users. The privilege-owner is also responsible for:

- Addition of Privilege to user.
- Deletion or Revocation of a user's Privilege.

Addition and revocation of privileges deal with modifying the Member List for a user. The Member List, as discussed below, contains the privileges a user is authorized to send with. A user can be added or revoked to the Member List only by the privilege-owner. If a user wishes to be added or deleted, a request should be sent to the privilege-owner. If the privilege-owner considers the request, the Member List will be updated at the P-Server; otherwise, the request is rejected.

However, if a user abuses the privilege, the privilege-owner should revoke the user. The revocation of the user will be performed by removing the user from the Member List. As the private key of the privilege is not revealed to the user, the privilege-owner need not create another PKI key pair for the privilege.

Privilege-List Maintenance

Each user maintains the Privilege List at the P-Verifier. This information needs to be updated by the user to classify the emails based on the privileges listed in the Privilege List. If a user wishes to honor a privilege, the user updates the privilege list with the P-Verifier. Adding a privilege to the privilege list is similar to maintaining white-lists albeit at the server side.

To assist users with initial list of privileges, a default list of privileges can be assigned to users by the

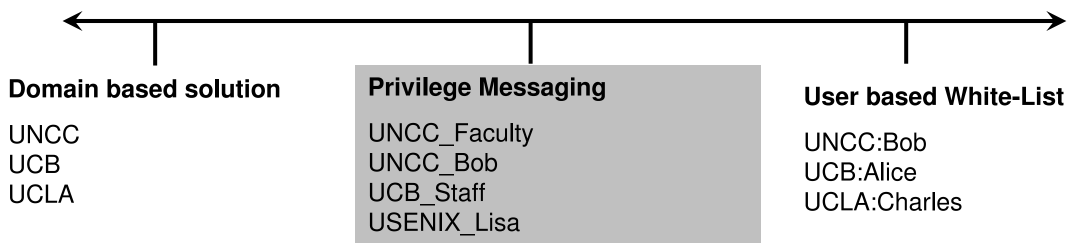


Figure 6: Example credentials maintained in different technologies.

mail service provider. This allows a default privileges associated with a user during the setup phase. In the absence of a user's input, which we believe quite common, the service provider's default list will be used. Some user-profiling and personalization techniques may be useful in determining the list on behalf of the user.

P-Messaging: Design and Benefits

This section discusses the various design decisions for Privilege Messaging. Using the CoT, Privilege Messaging provides a QoS over the current email architecture. The additional benefits are that P-Messaging would allow fine-granular privilege maintenance which allows the negative reputation to be contained within a privilege rather than the domain. Having the privilege in the white-list as compared to individual email IDs allows smaller list to be maintained. Also, a new correspondent will not be considered unsolicited sender.

Fine-Granular Reputation Management Than the Domain-Based Solutions

As discussed in the related work, P-Messaging is a solution falling between domain-based solutions and the user based white-lists in terms of its granularity. Figure 6 shows examples of the credentials for the different technologies. In the case of the domain-based solutions, the unit of credential is a domain. For P-Messaging, the credential is maintained per P-Tag, which can contain either single user or a group of users.

A domain-based solution publishes the credentials in the DNS records; therefore each mail sent has a single credential for the entire domain. With domain-based solutions, when a domain is registered it will be given a full authorization to send a message to other domains. However, when P-Server is registered, the domain will be given an authorization to issue and manage the P-Tag, not the right (authorization) to send the message. Privilege Messaging can be installed over multiple P-Servers on a domain where each P-Server maintains multiple privileges. Having multiple privileges allows negative reputations to be contained to a single privilege rather than a complete privilege server.

DKIM allows public keys to be created and embedded into the DNS records by the mail server itself, whereas Privilege Messaging requires the P-Server to publish its public key with a CA, the P-Messaging Trust Authority. Due to CoT, the trust on a sender is verified before the message is accepted at the receiver. In other domain-based solutions, the message is accepted without verification by a trusted third party.

In P-Messaging, the negative reputation is constrained into a single privilege as compared to the complete domain. Furthermore, P-Messaging provides QoS by automatically classifying the emails. This is further discussed in the next subsection.

Privilege Messaging has multiple keys that are required to classify emails. As discussed above, P-Server and its privileges are associated with their own corresponding PKI key pair. In case of the loss of the privilege key, the privilege-owner can request the administrator to reissue a new PKI pair for the privilege. If the P-Server's key is lost, the administrator requests the P-Messaging Trust Authority with a new PKI key pair.

Maintainable White-list Using Privilege-ID Rather Than Individual Email ID

The credentials for the user-based white-lists, as shown in Figure 6, are the individual email IDs. As discussed earlier, the privileges are the credentials in Privilege Messaging. In comparison to user-based white lists, the mails are not classified based on the sender's privilege or an email contents in Privilege Messaging.

The deployment of P-Messaging is valuable to an organization since the user key maintenance [16] is eliminated by maintaining the keys on an infrastructure level. It needs to be noted that PGP can still be used in conjunction with P-Messaging on a user-level basis for providing confidential and integrity services for an individual.

With white-lists, a new correspondent might be classified as an unsolicited sender. The benefit of P-Messaging is that a new correspondent may not be classified as an unsolicited sender.

Automatic Email Classification Based on the P-Tag

Based on the privileges presented, the emails are classified automatically based on the privileges that are accepted by the users. The automatic classification of emails, depending on the sender privileges and those that are honored by the receiver, provides the ability to classify the mails into three classes: the privileged class, underprivileged class and no-privilege class. The classification provides the QoS by allowing similar emails to be checked in one location.

In addition to such automatic classification, email clients can create multiple 'inboxes' based on accepted P-Tags. For example, the end-user email client can move a message signed by USENIX privilege into USENIX folder based on a user-specified rule. Email clients can be programmed to create separate inboxes for each privilege subscribed. This allows the server-side classification and user-side display of the classified emails.

Retaining the Benefits of Existing Email Architecture

In this section, we discuss the benefits of the current email architecture that are also supported by Privilege Messaging. Privilege Messaging provides sender privacy, incremental deployment over the current email infrastructure and use for the large-scale email service providers.

Sender Privacy

As an additional benefit, P-Messaging provides sender privacy. Once the user is authenticated at the sending P-Server, the sender information can be removed from the email, without the loss of the privileges associated with the email. Similarly, relaying emails from domains is possible. The privilege-signed email without the header can then be verified and classified based on the privileges. With this ability, users can send email to a person in the group without their information. For instance, the sender information from a student's email in a class can be removed, and the email be delivered to the professor verifying that the email is coming from the class, but not from a specific student. Though the message is sent without the sender information, the message will not be considered as unsolicited using Privilege Messaging.

Certain websites, such as job search service websites, send a notification email to the registered clients. However, due to web-site policy, the sender email ID is changed to the receiver's email ID. Such valid source privacy can be provided by P-Messaging. Sender information can be removed from the emails, yet classification can be easily performed without them being marked as unsolicited.

Incremental Deployment

Privilege Messaging can be deployed incrementally over the current email infrastructure. Through P-Messaging, the unsigned messages will be classified into the no-privilege class. To classify emails from the users into other privilege classes, P-Messaging should be deployed by the sender and the receiver. P-Messaging complements the existing email infrastructure, retaining all the benefits (e.g., relaying) at the same time adding the much-needed authorization framework.

With the large scale adoption of Privilege Messaging, the unsolicited mails sent over the wire will be reduced, allowing the mail to be sent only from verifiable servers. And while much of the transmitted bulk emails would still be mostly unwanted, the source of the sender can be verified and the ability to restrict those emails is also provided to the user.

Large-Scale Email Service Providers

Privilege Messaging allows system administrators to deploy P-Messaging for large scale email providers that have users who abuse the system. The providers can classify their user base into multiple privileges; for example, on the number of years the email ID has been in use and/or the location of the user. This creates smaller subsets of users to be dealt with. If a user abuses their privileges, the email providers can revoke the user and their privileges. Alternatively, the users who wish to use their email accounts with P-Messaging can be provided with 'Signing Services,' which sign the users' email on behalf of the user. Meanwhile, the P-Server that sends the mails can request additional terms with P-Messaging Trust Authority, so that a few negative instances would not revoke the P-Server from the CoT.

P-Messaging Prototype Implementation and Configuration

In this section, we present the implementation of P-Messaging. We discuss the sender and the receiver configuration, and demonstrate the processes involved in setting up P-Messaging and the process for sending and receiving emails.

Implementation Details

P-Messaging has been implemented using Java 1.4. The P-Messaging Trust Authority uses Remote Method Invocation (RMI) to generate the PKI key pair for the privileges as well as for the P-Server. For transmitting the mail from the client to the P-Server, the RMI architecture is used.

The P-Server uses Java Mail API to transmit the mail to the MTA. The MTA is Sendmail. Sendmail has been chosen as it allows an external entity, a Milter, to classify the messages. A java implementation of Milter is provided using Jilter API [19]. The systems running the mail servers also provide IMAP services using the Cyrus IMAP server. This module essentially functions as the P-Verifier that provides digital signature verification.

Sender and Receiver Configurations

In this section, we discuss the changes to the sender and the receiver side for deploying P-Messaging at an organization. Privilege Messaging allows two different changes to the configurations to the sender and requires a single configuration to the receiver side. In order to use P-Messing with legacy email servers, the email clients need an added mechanism that shows the listing of privileges at the time of sending. The retrieved emails need to be classified based on the associated P-Tag.

Sender Configuration

As described above, the sender side configuration can be created by two different methods. The first method requires changes to the email client so that the user privileges can be retrieved from the P-Server. With the change in the email client, the user can select the privilege that the email needs to be sent with. The second configuration requires the privilege selection with the help of a simple rule based privilege selection engine where a privilege is selected while sending an email to a specific user/ group of users.

Receiver Configuration

As discussed above, the receiver side configuration is minimal. The receiver side installation is a one time installation of P-Verifier. The P-Verifier, which is a Jilter, needs to be added to the Sendmail configuration file. The following is the configuration lines:

```
O InputMailFilters = <Jilter Name>
X<Jilter Name>, S=inet:<port>@<IP address>
```

The configuration needs to be added to the /etc/mail/sendmail.cf (Fedora Core) configuration file. Figure 7 shows the configuration for Sendmail.

Test Setup

All of the experiments have been performed using the following devices and networks with the specified configurations. The client and the P-Messaging Trust Authority runs on Intel Pentium 4 CPU 3.20

GHz with 1.5 GB RAM running Microsoft Windows XP Professional version 2002.

The two mail servers run Sendmail 8.12.10. The first system is: Intel Pentium 4 CPU 2.5 GHz with 512 MB RAM running Linux 2.6.14 Kernel: this system accepts mails. The second system is an Intel Celeron 2.5 GHz with 1 GB RAM running Linux 2.6.12.6. This system serves as the primary P-Server, the sender domain. The Local Area Network bandwidth was about 100 Mbps with a delay of about 0.1-0.2 milliseconds.

Adding and Revoking a P-Server to the CoT

As discussed in the above sections, maintenance of the CoT is important for a P-Server to place trust on any other entity. Figure 8 shows the process of adding a P-Server to the CoT. The process of adding the P-Server to the CoT involves creation of a PKI key pair. When installing a P-Server, the P-Server asks the necessary questions to create a PKI key pair. Once the information is gathered, this information is sent to the P-Messaging Trust Authority over RMI which creates the key pair.

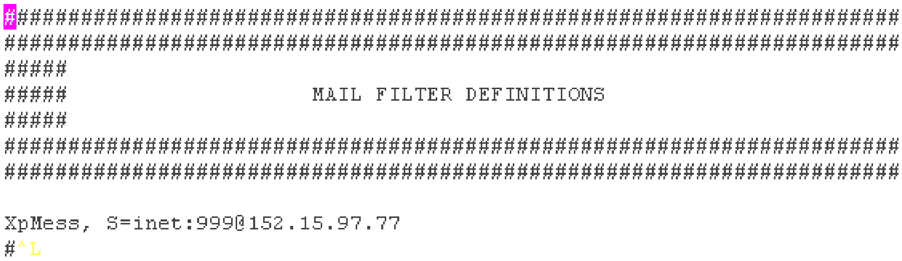
Another important aspect for maintaining the CoT is the revocation of a P-Server. The process of revocation is carried out at the P-Messaging Trust Authority. The revocation is performed by removing the P-Server from the trusted list. Figure 9 shows the revocation process of the P-Server. The present prototype implementation of Privilege Messaging does not cache the public keys of the peer P-Server, the verification is done by looking up the P-Messaging Trust Authority for the privilege's public key. Thus, the present version of P-Messaging does not use Certificate Revocation List (CRL) to remove the defaulting P-Server.

Maintenance of the Privilege

As discussed above, each privilege is created by the P-Server administrator and is managed by a privilege-owner. The privilege-owner is capable of creating and deleting a privilege. Once the privilege modifier is started, a user can create a privilege and assign a privilege-owner. Privilege creation involves the creation of a PKI key pair with the predefined site information. Once a privilege is created, users can be added to the privilege's member list. Figure 10 shows the privilege management.

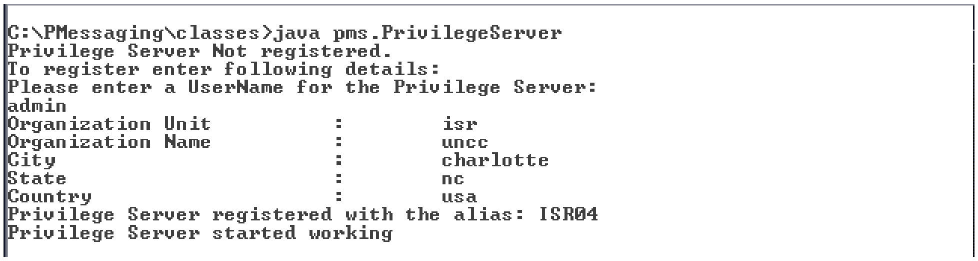
Privilege List Maintenance

Apart from maintaining the Member-list, a user needs to maintain the Privilege List. The Privilege List is a list maintained at the P-Verifier by each user. The list contains all the privileges that are accepted by the



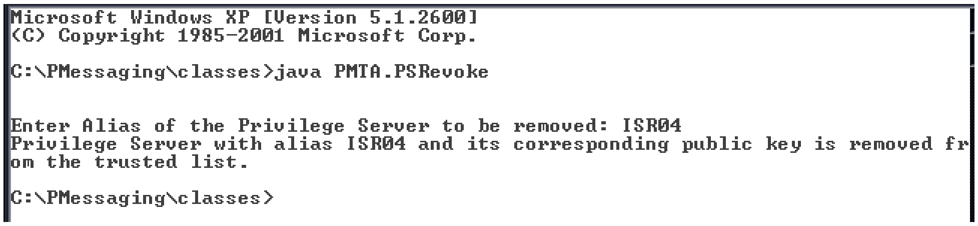
```
######################################################################
######################################################################
#####
#####                   MAIL FILTER DEFINITIONS
#####
######################################################################
######################################################################

XpMess, S=inet:999@152.15.97.77
#^L
```

Figure 7: Jilter Configuration (P-Verifier) at the receiver domain.



```
C:\PMessaging\classes>java pms.PrivilegeServer
Privilege Server Not registered.
To register enter following details:
Please enter a UserName for the Privilege Server:
admin
Organization Unit          :        isr
Organization Name          :        uncc
City                       :        charlotte
State                      :        nc
Country                    :        usa
Privilege Server registered with the alias: ISR04
Privilege Server started working
```

Figure 8: Registration of Privilege Server with P-Messaging Trust Authority which generates Public key for Privilege Server.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\PMessaging\classes>java PMTA.PSRevoke


Enter Alias of the Privilege Server to be removed: ISR04
Privilege Server with alias ISR04 and its corresponding public key is removed fr
om the trusted list.

C:\PMessaging\classes>
```

Figure 9: Revocation of Privilege Server at P-Messaging Trust Authority. The Public key cannot be retrieved by peer Privilege Servers after revocation.

user. Figure 11 shows the interface where a user can honor or dishonor a privilege from the privilege list. The privilege list maintenance is implemented using RMI.

Apart from privilege creation, the P-Server administrator should be able to revoke a privilege from the list. Revocation of a privilege involves removing the users from the privilege and modifying the Member-list. The PKI key pair is invalidated so that the users can no longer use the privilege.

Sending a Privileged Email

Figure 12 shows the method in which a mail is sent by a user using our initial prototype. Once the user selects 'Send Email' from the client interface, the interface shows all the privileges that can be used to send an email with. Once the privilege is selected, the message is then sent to the P-Server. The P-Server adds the privilege signature to the email and sends the email out.

Classification of the Email By P-Verifier

Figure 13 shows email classification by P-Verifier. As described in previous sections, P-Verifier is a Jilter interface that verifies the emails based on the privilege information provided by the email.

P-Verifier classifies the mail, as shown in the Figure 13; the Jilter accepts the emails on a specified IP address and the port. The Output is shown in the following format:

```
<Receiver> <Sender> <Subject>:
  <Privilege Verification Status>
```

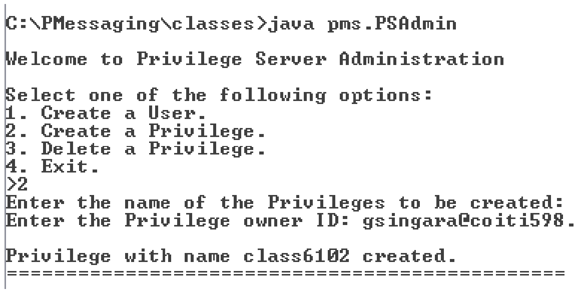
Retrieval of Email By an Email Client

Figure 14 shows the mechanism used to retrieve the mail. The email client is a prototype model that demonstrates the classification of emails that are classified by the P-Verifier. Once the emails are retrieved, the messages are classified based on the attached P-Tag.

An email client that creates multiple 'inboxes' for each privilege for a user would enable access to the emails in one location. Such a service would provide QoS for the users who would like to access all their emails at one location.

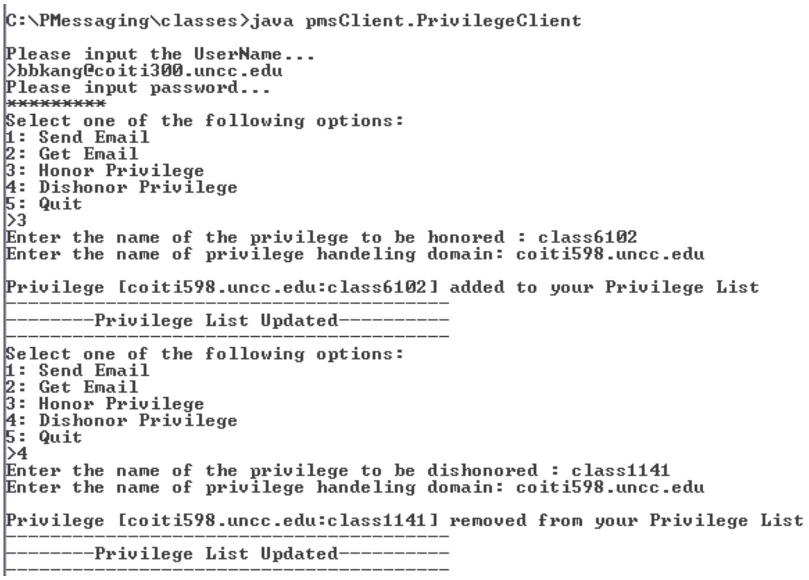
Performance Results

This section demonstrates the performance of P-Messaging. The results were performed on the same configurations as described earlier.



```
C:\PMessaging\classes>java pms.PSAdmin

Welcome to Privilege Server Administration

Select one of the following options:
1. Create a User.
2. Create a Privilege.
3. Delete a Privilege.
4. Exit.
>2
Enter the name of the Privileges to be created: class6102
Enter the Privilege owner ID: gsingara@coiti598.uncc.edu

Privilege with name class6102 created.
===============================================
```

Figure 10: Privilege Server Manager Interface that connects to PS Admin service to create and delete a privilege.



```
C:\PMessaging\classes>java pmsClient.PrivilegeClient

Please input the UserName...
>bbkang@coiti300.uncc.edu
Please input password...
*********
Select one of the following options:
1: Send Email
2: Get Email
3: Honor Privilege
4: Dishonor Privilege
5: Quit
>3
Enter the name of the privilege to be honored : class6102
Enter the name of privilege handeling domain: coiti598.uncc.edu

Privilege [coiti598.uncc.edu:class6102] added to your Privilege List
----------------------------------------
--------Privilege List Updated----------
----------------------------------------
Select one of the following options:
1: Send Email
2: Get Email
3: Honor Privilege
4: Dishonor Privilege
5: Quit
>4
Enter the name of the privilege to be dishonored : class1141
Enter the name of privilege handeling domain: coiti598.uncc.edu

Privilege [coiti598.uncc.edu:class1141] removed from your Privilege List
----------------------------------------
--------Privilege List Updated----------
----------------------------------------
```

Figure 11: Client Interface that connects to Privilege Server to honor or dishonor a privilege for a user. To honor or dishonor a privilege at client, the information about the Privilege Server that handles the privilege is required. The example above shows a professor managing two different privileges: class1141 and class6102.

Our experiments were geared towards demonstrating the performance costs associated with Privilege Messaging compared with PGP-signed email. To determine the P-Messaging performance, we conducted the following experiments:

1. P-Tag Generation Time
2. P-Tag Verification Time
3. Privilege Generation and Verification Time

P-Tag Generation Results

To demonstrate the overhead incurred due to generation of P-Tags, we compared P-Messaging's tag generation performance with the time taken to generate PGP digital signature and unsigned emails. Figure 15 shows our results. The time taken to generate the tag was reasonably higher PGP and unsigned messages, the overhead grows linearly. The overhead includes the time taken to request for a privilege using RMI and generation of two signatures by the P-Server. The result that is shown in the Figure 15 is expected as the P-Messaging performs a double signature and takes twice the time as compared to the time taken to generate PGP signature.

P-Tag Verification Results

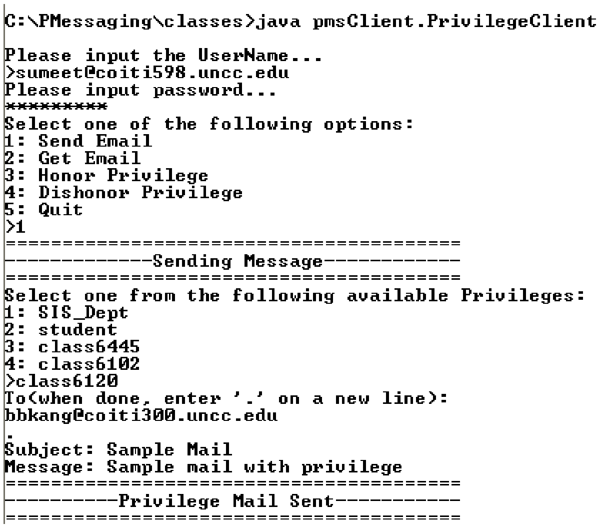
To demonstrate the overhead incurred due to verification of the privilege, we compared the time taken to verify the Privilege with time taken to verify a mail using PGP. Our results are shown in Figure 16 where the time taken to verify the emails is twice the time taken to verify the PGP signed mail. The results are as expected since the Privilege signed mails include two signatures as compared to one signature in PGP.

Privilege Generation and Verification Time

Our experiments show that the time taken to generate a Privilege-tag for an email and send it over LAN was about 0.16 sec. This time included the time taken to generate double signatures: one for the privilege and another for the P-Server server. The time taken to verify a message was about 0.09 Sec, again this time involved the time taken to verifying the P-Server signature, and then the Privilege signature. It also involved the time to retrieve the privileges' public key from the sender's P-Server.

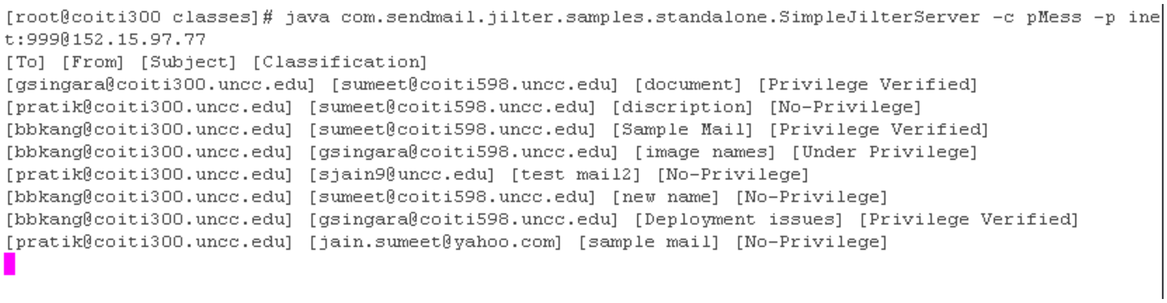
Future Work

This section discusses the future work for Privilege Messaging. For P-Messaging to be effective, we further need a mechanism that would allow the receiver to evaluate the Privilege-tags along with the corresponding P-Server. In order to prevent a privilege from being used for sending unsolicited content, a



```
C:\PMessaging\classes>java pmsClient.PrivilegeClient

Please input the UserName...
>sumeet@coiti598.uncc.edu
Please input password...
*********
Select one of the following options:
1: Send Email
2: Get Email
3: Honor Privilege
4: Dishonor Privilege
5: Quit
>1
=======================================
-------------Sending Message-------------
=======================================
Select one from the following available Privileges:
1: SIS_Dept
2: student
3: class6445
4: class6102
>class6120
To(when done, enter '.' on a new line):
bbkang@coiti300.uncc.edu
.
Subject: Sample Mail
Message: Sample mail with privilege
=======================================
----------Privilege Mail Sent-----------
=======================================
```

Figure 12: Client interface to connect to a Privilege Server to send an email. The client required to select a privilege to send an email.



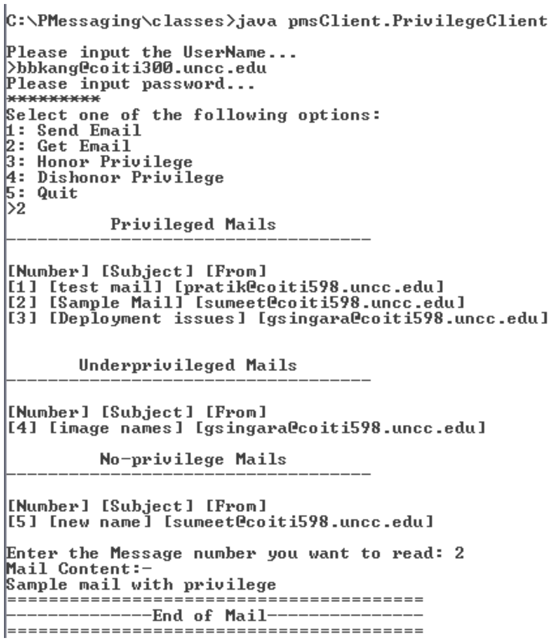
```
[root@coiti300 classes]# java com.sendmail.jilter.samples.standalone.SimpleJilterServer -c pMess -p ine
t:999@152.15.97.77
[To] [From] [Subject] [Classification]
[gsingara@coiti300.uncc.edu] [sumeet@coiti598.uncc.edu] [document] [Privilege Verified]
[pratik@coiti300.uncc.edu] [sumeet@coiti598.uncc.edu] [discription] [No-Privilege]
[bbkang@coiti300.uncc.edu] [sumeet@coiti598.uncc.edu] [Sample Mail] [Privilege Verified]
[bbkang@coiti300.uncc.edu] [gsingara@coiti598.uncc.edu] [image names] [Under Privilege]
[pratik@coiti300.uncc.edu] [sjain9@uncc.edu] [test mail2] [No-Privilege]
[bbkang@coiti300.uncc.edu] [sumeet@coiti598.uncc.edu] [new name] [No-Privilege]
[bbkang@coiti300.uncc.edu] [gsingara@coiti598.uncc.edu] [Deployment issues] [Privilege Verified]
[pratik@coiti300.uncc.edu] [jain.sumeet@yahoo.com] [sample mail] [No-Privilege]
```

Figure 13: Demonstration of P-Verifier interface in a verbose mode for the classification of emails at the receiver domain.

reputation system needs to be developed so that the reputation values of P-servers and their privileges are periodically updated with an increase for right behavior and a decrease for negative behavior. Such a reputation system in a distributed environment with partially trusted entities is difficult to achieve, since each partially-trusted server might hold varying number of privileges each with varying number of users. The negative reputation of a privilege can propagate to the server and therefore the reputation of all privileges associated might be affected. Hence, an effective reputation management would be essential for the successful adoption of P-Messaging. The reputation value should be embedded into a certificate for the P-Server and the Privilege-tag.

Privilege Management requires better interfaces for Privilege-tag management. A usage-study on the ease of P-Messaging usage would allow a better UI design or additional mechanisms for Privilege Management. For instance, consider a case when the P-Tag owner receives a request for addition to the privilege. The P-Tag owner will give access only to those users who can successfully provide their identity: the requestors should themselves be a part of some verifiable Privilege Server, thereby reducing the number of requests received by the owner.

Privilege selection is an important aspect to be performed by the senders. For a message to be created in a receiver's mailbox, P-Messaging mandates that the sender should have at least one privilege that the receiver would accept. A protocol should exist that pre-computes a privilege that a sender would require so that a message can be accepted at the receiver. This protocol can perform a data mining on the privileges



```
C:\PMessaging\classes>java pmsClient.PrivilegeClient

Please input the UserName...
>bbkang@coiti300.uncc.edu
Please input password...
*********
Select one of the following options:
1: Send Email
2: Get Email
3: Honor Privilege
4: Dishonor Privilege
5: Quit
>2
            Privileged Mails
-----------------------------------

[Number] [Subject] [From]
[1] [test mail] [pratik@coiti598.uncc.edu]
[2] [Sample Mail] [sumeet@coiti598.uncc.edu]
[3] [Deployment issues] [gsingara@coiti598.uncc.edu]


       Underprivileged Mails
-----------------------------------

[Number] [Subject] [From]
[4] [image names] [gsingara@coiti598.uncc.edu]

          No-privilege Mails
-----------------------------------

[Number] [Subject] [From]
[5] [new name] [sumeet@coiti598.uncc.edu]

Enter the Message number you want to read: 2
Mail Content:-
Sample mail with privilege
========================================
--------------End of Mail---------------
========================================
```

Figure 14: Prototype client implementation allowing user to retrieve classified emails.

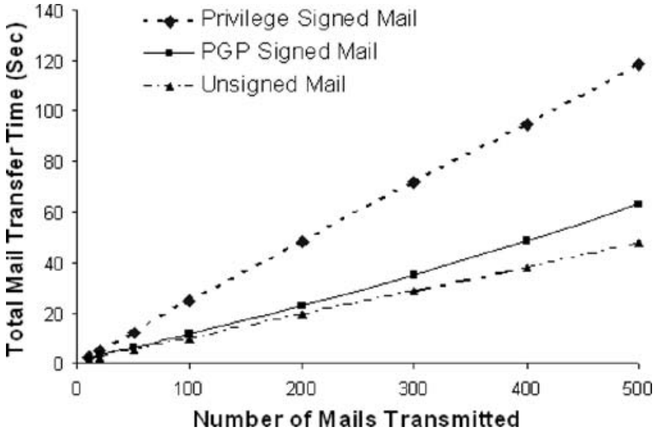


Figure 15: Comparing time taken for sending P-Tag attached emails, emails with PGP signature and unsigned emails.

of the sender and the receiver to select a common privilege. Data mining can be useful for recursive-privilege mechanism where the senders can add additional credentials, which they hold on peer P-Server. The additional credentials are created based on the privileges that the receiver honors.

The present architecture does not cache public keys for a P-Server and its privileges. Caching the public keys allows faster verification of emails. Introducing key caching requires Certificate Revocation List (CRL) management. CRL management would be challenging in a distributed architecture for revoking the certificate of a privilege maintained by peer P-Server. A better privilege verification would consider reducing the number of round trips required to verify the privilege. Also, previously sent emails need to be verified in the event where the Privilege's key has been revoked.

Conclusion

In this paper, we presented an authorization framework, called Privilege Messaging, overlaying the existing email infrastructure while retaining the beneficial aspects such as relaying. For the sender, P-Messaging provides a mechanism that allows email delivery only if the sender possesses the privilege that the receiver would accept. Based on the email privileges, the email is classified automatically according to the Privilege Tag at the receiver, providing QoS for the user. Having the privileges in the white-list as compared to individual email IDs allows a smaller list to be maintained and a new correspondent may not be regarded as an unsolicited.

Each Privilege Server manages multiple privileges as opposed to a single credential as previously proposed in domain-based authentication schemes. In case of the compromise or spam being propagated from one domain, the negative reputation is contained within a privilege rather than the complete domain.

With the help of P-Messaging, the email's authenticity is verified by a trusted third party. To allow privileges to be verified across domains, P-Messaging establishes a Circle of Trust among the P-Server for privilege verification. P-Messaging performs dual digital signature on an email, first by the assigned privilege and then by P-Server, allowing peers in the CoT to verify the email's authenticity. This ensures that only authorized users can send messages only if their P-Server is a member of CoT, and that a P-Server needs to limit the unwanted email that it transmits or it would be revoked from the CoT.

Privilege Messaging can be deployed incrementally; P-Messaging is a gradual process of introducing the authorization over the current email infrastructure. To support such deployment, P-Messaging can coexist with other technologies, providing trust-based email service over MTA. P-Messaging is designed to work well over the existing SMTP infrastructure with minimal user-level interaction and deployment overhead for the authorization provided.

Finally, for more information, please visit: http://isr.uncc.edu/pmessaging.

Acknowledgment

We would like to thank our shepherd, Rowan Littell, and the anonymous reviewers for their insightful comments. We would also like to show our appreciation to our LISA copy-editor, Rob Kolstad, for his excellent service. Finally, we thank our ISR lab members for their help from the early stage of this paper.

Author Biographies

Brent Hoon Kang received his Ph.D in Computer Science from the University of California at Berkeley, working on the Berkeley Digital Library and OceanStore project. Prior to Berkeley, he received an M.S in Computer Science from the University of Maryland at College Park, and a B.S in Computer Science and Statistics from Seoul National University with 1st place distinction among computer science majors. Since Fall 2004, he has been an assistant professor at the University of North Carolina (UNC) at

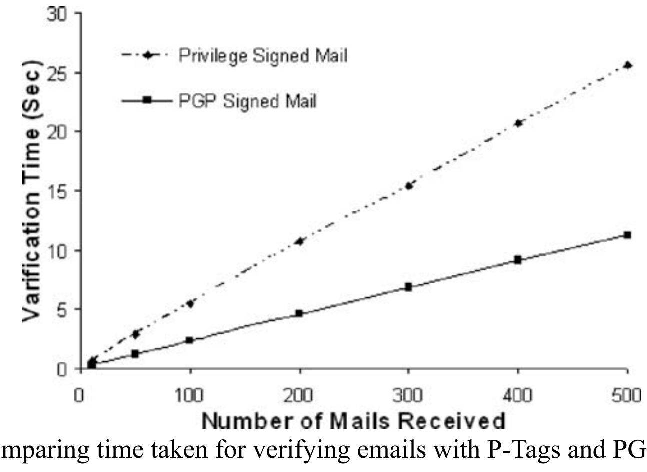


Figure 16: Comparing time taken for verifying emails with P-Tags and PGP signed emails.

Charlotte. He is currently leading the Infrastructure Systems Research (ISR) Lab with a focus on securely architecting large-scale infrastructure systems, working with five graduate students. Hoon can be reached at bbkang@uncc.edu .

Gautam Singaraju is a fourth year doctoral student advised by Dr. Kang. He completed an M.S in Computer Science at UNC Charlotte in 2003. Since then, he has also been a volunteer System Administrator for a global non-profit organization. Gautam can be reached at gsingara@uncc.edu .

Sumeet Jain is a second year graduate student working on his master degree with Dr. Kang at UNC Charlotte. He completed an M.S. in Computer Science at Rajiv Gandhi Technical University, India, and worked with Choksi Laboratories Limited as a Software Developer. Sumeet can be reached at sjain9@uncc.edu .

Bibliography

[1] Ahmed, S., F. Mithun, "Word stemming to enhance spam filtering," *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[2] Allman, E., *DomainKeys Identified Mail (DKIM): Introduction and Overview*, 2005, http://mipassoc.org/dkim/info/DKIM-Intro-Allman.html .

[3] Andreolini, M., M. Colajanni, F. Mazzoni, L. Messori, "HoneySpam: Honeypots fighting spam at the source," *Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop*, Cambridge, 2005.

[4] *CAN-SPAM Act: Requirements for Commercial Emailers*, http://www.ftc.gov/bcp/conline/pubs/buspubs/canspam.htm .

[5] Duan, Z., K. Gopalan, Y. Dong, "Push vs. Pull: Implications of Protocol Design on Controlling Unwanted Traffic," *Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop*, Cambridge, 2005.

[6] Finnegan, O., *Email Deliverability Getting your Email into the inbox*, 2005, http://www.ieinternet.com/mailwall/Email_Deliverability_whitepaper.pdf .

[7] Gomes, L. H., C. Cazita, J. M. Almeida, V. Almeida, W. Meira, "Characterizing spam traffic," *Proc. 4th ACM SIGCOMM Conference on Internet Measurement*, 2004.

[8] Gray, A., M. Haahr, "Personalised, Collaborative spam filtering," *Proceedings the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[9] Hardy, I. R., *The Evolution of ARPANET Email*, Thesis, Department of History, University of California, 1996.

[10] Kolcz, A., A. Chowdhury, J. Alspector, "The impact of feature selection on signature-driven spam detection," *Proc. the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[11] Leiba, B., N. Borenstein, "A multifaceted approach to spam reduction," *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[12] Microsoft Corporation, *Sender ID Framework – Executive Overview*, 2004.

[13] Milletary, J., *Technical Trends in Phishing Attacks*, 2006, http://www.cert.org/archive/pdf/Phishing_trends.pdf .

[14] NACHA, "Phishing losses total $500 million," Technical report, NACHA – The Electronic Payments Association, 2004.

[15] Neustaedter, C., A. J. Bernheim Brush, Marc A. Smith, Danyel Fisher, "The Social Network and Relationship Finder: Social Sorting for Email Triage," *Proc. Conference on Email and Anti-Spam (CEAS)*, 2005.

[16] Price, W., *Inside PGP Key Reconstruction*, A PGP corporation White paper, 2003.

[17] Realtime Blackhole List, Mail Abuse Prevention System LLC, California, 2002, http://www.mail-abuse.org/rbl/.

[18] Segal, R., J. Crawford, J. Kephart, B. Leiba, "Spamguru: An enterprise anti-spam filtering system," *Proc. of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[19] *Sendmail-Jilter API*, 2005, http://sendmail-jilter.sourceforge.net/index.html .

[20] W. Wong, M., *Sender Authentication: What to do*, Technical Document, 2004, http://www.openspf.org/whitepaper.pdf .

[21] Yahoo Inc., *DomainKeys: Proving and Protecting Email Sender Identity*, http://antispam.yahoo.com/domainkeys .

[22] Zimmermann, P., *The Official PGP User's Guide*, MIT Press, Cambridge, 1995.