情報処理学会
グループウェアとネットワークサービス研究会
2004年11月11日，12日

電子メールの送信元認証におけるハンドシェーク活用の提案

光田　智史† 　相田　仁†

† 東京大学大学院 新領域創成科学研究科
〒277–8561 千葉県柏市柏の葉 5–1–5 基盤棟 611
E-mail: †mitsuda@aida.k.u-tokyo.ac.jp, †aida@k.u-tokyo.ac.jp

あらまし　近年 SPAM メールが増え続けている理由の 1 つは，SMTP においてはメールの受信側が送信側を信頼することが前提になっており，送られてきたメールの送信元を確かめる手段がないことであり，送信元認証に関する研究が盛んに行われている．そこで本稿では送信側のサーバが送ったメールの情報を保持しておき，受信側のメールサーバは送信側のメールサーバに対し，メールを送ったことを問い合わせて確認してから受信するメール送信システムを提案する．

キーワード　電子メール，SPAM，送信元認証，ハンドシェーク

Handshake Approach for Sender Authentication in Email System

Satoshi MITSUDA† and Hitoshi AIDA†

† Graduate School of Frontier Sciences, The University of Tokyo
611 Kibantou, 5–1–5 Kashiwanoha, Kashiwa-shi, Chiba, 277–8561 Japan
E-mail: †mitsuda@aida.k.u-tokyo.ac.jp, †aida@k.u-tokyo.ac.jp

Abstract　The number of SPAM has increased over the past few years. Most of them lie about their identity due to lack of sender authentication in SMTP. Thus, we propose a new mail transfer system using handshake method. In this system, Sender MTA store the checksum of mail in a database, and receiver MTA verify the authenticity of the mail to sender domain.

Key words　Email, SPAM, Sender Authentication, Handshake

1. はじめに

電子メールは，インターネットにおいて WWW と共に最も普及したサービスの一つであり，もはや人々の生活になくてはならないものとなっている．しかしその電子メールが広く利用されるにつれ，SPAM メールが深刻な社会問題となってきている．このように SPAM メールが横行している原因は SMTP(Simple Mail Transfer Protocol) という送信プロトコルにあると言われている．SMTP においてはメールの受信側が，送信側を信頼するというモデルの下に成り立っており，送信元のメールアドレスが偽装できてしまうなどの問題がある．そこで本稿では，メールを送信する際に，受信側のメールサーバが送信側のメールサーバに対し，メールを送ったことを確認してから受信するメール送信システムを提案する．

2. SPAM メール

2.1 SPAM メールとその影響

SPAM メールは電子メールの普及と共に増え続け，世界規模で見ると 2003 年についに全メールのうち SPAM メールが占める割合が 50%を超えたと言われている [1]．今現在はインターネット先進国であるアメリカなどの方が，より深刻な問題になっているが，日本でも携帯電話の SPAM メールがユーザを困らせ，様々な社会問題を引き起こしている．SPAM メールは，送信者側から見ると，広告の手段として他のタイプのダイレクト・マーケティングよりも短時間かつ低コストで送信できるため，返答率は低いとしてもはるかに効果のあるものである．一方受信者側は，よけいな通信費やリソースを浪費し，その対応に貴重な時間とコストを費やさざるを得ない．

最近では，実在する企業に似せたホームページを作り，そこにリンクを張った電子メールを送信し，ユーザを誘導して，クレジットカードの番号などの個人情報を盗み取るといった “phishing” という詐欺の被害が非常に増えている．そして増え続ける SPAM メールによって電子メールが信用できなくなった人もかなりの割合に達してきている [2].

このように SPAM メールは多くの問題を抱え，数々の研究や議論がなされている [3]〜[6]．ただ，SPAM メールと一言に言っても，その種類も送られる手段も様々であるため，それを防止する決定的な手法はなく，様々な分野からのアプローチが必要である．

まず政治面において，SPAM メールを取り締まる法律の制定が各国でなされている．アメリカでは 2004 年 1 月に CAN-SPAM(Controlling the Assault of Non-Solicited Pornography and Marketing) 法が施行された [7]．この法律においては，広告メールに返信用のアドレスや，メール配信を中止する Web ページへのリンクをつけることを義務付け，メールヘッダの偽造などを禁止している．日本においても 2002 年に「特定商取引に関する法律の改正」及び「特定電子メールの送信の適正化等に関する法律」などが制定されている [8]．しかし，いずれもあまり効果があがっていない．SPAM メールは一国内に留まらず国境を越えて送られ続けるため，政府機関や企業の国際的な取り組みが不可欠である．近年 OECD，IETF などの機関が，国際的な SPAM 対策に乗り出しているが，未だに大きな効果が出ているとは言えない．

2.2 SMTP

現在のインターネットにおける電子メールでは SMTP というプロトコルが使われている．SMTP は，1982 年に RFC821 として公開されて以来様々な拡張がなされてきたが，基本的な送信の仕方は変わっていない．HELO(EHLO) コマンドで自分のホスト名を名乗り，MAIL FROM コマンドと RCPT TO コマンドでそれぞれ送信元と送信先のメールアドレスを示し，DATA コマンドでメールのヘッダを含めた本文を送信し，QUIT コマンドで終了するといったやりとりになる．

電子メールの配送の様子を Fig.1 に示す．MUA(Mail User Agent) から送信されたメールはメールサーバの MTA(Mail Transfer Agent) へ SMTP を用いて送られる．それを受け取った MTA は，宛て先メールアドレスのドメイン名から，DNS の MX レコードを使用して送信先 MTA を探し出し，SMTP で送信する(注1)．宛て先

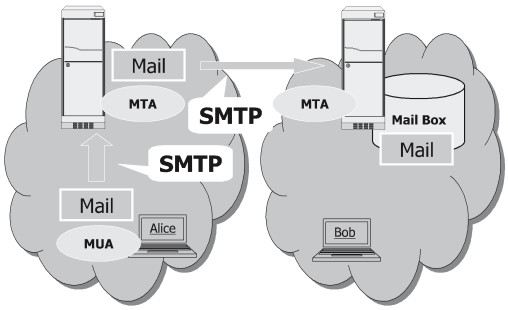

Fig. 1 Delivery of Email Messages.

MTA は，ユーザのメールボックスにメールを配信する．ユーザがメールサーバに届いたメールを手元の PC で見る際には，POP や IMAP といったプロトコルを使う．

ここで重要なことは，配送は SMTP コマンドの後のエンベロープアドレスに基づいて行われるということである．メールヘッダはメッセージデータの一部にすぎず，配送制御には使われない．そして，通常我々が目にするメールの差出人アドレスというのは，メールヘッダに書かれた “From:” フィールドを MUA が解釈して表示しているだけである．

2.3 SPAM メール増加の原因

SMTP が SPAM メールを助長させている理由は，メールの受信側が送信側を信頼することが前提になっており，送られてきたメールの送信元を確かめる手段がないことである．現状の SMTP では，MAIL FROM コマンドの後に続く Return-Path(注2)を偽ってメールを送っても，また Return-Path とメールヘッダの “From:” フィールドに何の関連性がなくても問題なく届いてしまう．すなわち，メールというシステムは送信側から受信側に向けて一方的に送信され，受信側はただそれを受け取るしかない “プッシュ型” のシステムであると言える．そして MUA で “From:” フィールドを見て差出人を調べても，それが本当にそのメールアドレスの人から来たのか確かめられないのである．

3. SPAM メール防止技術

SPAM メールを防止するための手法としては，以下のようなものが挙げられる．

a) RBL

MAPS [9] や ORDB [10] などの RBL(Realtime Blackhole List:不正中継ホストのデータベース) を参照して SPAM の可能性が高いメールを排除する．勝手に自分の

(注1)：送信先ホストが自分自身であった場合は，そのままメールボックスへ配信される

(注2)：MAIL FROM コマンドのアドレスはエラーメールなどを返すときなどに使われるためにこう呼ばれる．

サーバが登録されてメールが送信できなくなるなど課題も少なくない．

b） コンテンツフィルタリング

MUA の SPAM フィルタ機能などにより，SPAM メールによく見られるフレーズを検索し，削除する．ベイジアンフィルタなど，様々なアルゴリズムが盛んに研究されているが，あるアルゴリズムが普及すると，スパム業者はそれに対抗した手法を取るといったように，研究者とスパム業者のいたちごっこが依然として続いており，最近の SPAM メールは，知人を装ったり仕事の連絡を装ったりするなど手口が巧妙である．SPAM フィルタの一番の問題点は，フィルタリングの精度をを高くしようとすると SPAM でないメールが SPAM メールであると判断されてしまう危険性が高くなってしまうということである．

c） 電 子 署 名

SPAM メールの多くは送信元アドレスを偽装している．この偽装を検出することができれば，正規のメールか SPAM メールかの見分けがつく．送信元アドレスを確認する究極の方法は，PGP，S/MIME などのように公開鍵暗号方式を用いてメールに電子署名をつけることである．これらは，メッセージを秘密鍵で暗号化してから送信する．メールの受信者は送信者の公開鍵で復号することでそのメッセージを作成した人を確認することが出来き，End-to-End でメッセージの完全性が保証される．しかし，PGP は不特定多数の人とやりとりするような一般的な電子メールには向かず，また S/MIME は公開鍵を認証する認証局を用意する必要がある．暗号を用いる方法は，とても複雑でエラーが起こりやすいこと，PKI(公開鍵基盤) を整備し，1 人 1 人のユーザに導入する必要があること，秘密鍵が盗まれたことを検出しにくいことなどから近い将来に完全に実現される可能性は極めて少ないと考えられている．

d） ドメインレベルでの送信元認証

最近になって，送信元アドレスをドメインレベルで認証するプロトコルが IETF の MARID ワーキンググループなどで盛んに議論されている．現状の SMTP では，メールを “受信する”MTA の IP アドレスだけを DNS の MX レコードに書いておくが，これらはさらにメールを “送信する”MTA の情報も DNS に登録しておくことで，メールを受信したサーバが，送信元ドメインの権限のある IP アドレスからメールが送られてきているかを調べることを可能にしている．これらの目的とするところは，送信元 MTA(SMTP クライアント) を IP アドレスで認証することであるため，IP アドレスの偽装や DNS Spoofing までは対応できない．

2004 年に入って主に研究されてきた技術は，SPF(Sender Policy Framework) [11] と Caller ID for E-mail である．

SPF は America Online(AOL) が支持する技術で，受信側の MTA は送信元のエンベロープアドレスである Return-Path のドメイン名と SMTP クライアントの IP アドレスの対応を取る．例えば DNS に

```
example.com. TXT "v=spf1 mx -all"
```

と書いた場合，SMTP クライアントの IP アドレスが，example.com の MX レコードのホストの IP アドレスリストの中に存在したら SPF 認証を通過することになる．しかし，通常 “.forward” などによって転送するメールにおいては，Return-Path は変えられずに送られるため，転送する MTA の IP アドレスと Return-Path の不整合が生じてしまうという問題がある．

一方 Caller ID for E-mail は Microsoft が提唱している技術で SPF と異なり，ヘッダの送信元アドレスを認証する．Caller ID では，DNS に XML 形式で情報を発行するため，より複雑な処理が可能となっている．

両者は 2004 年 6 月に統合されて，Sender ID [12] という 1 つの仕様として IETF に提出された．しかし特許問題などで標準化が難航し，最近になって AOL が Sender ID の支持を撤回し，元の SPF を推進することに方針を変更した．これにより Sender ID の進展が見込めなくなり，IETF の Sender ID 作業部会は解散してしまい，普及にはまた当分の時間がかかると予想される．

また別の送信元認証として，DomainKeys という電子署名を使った技術も Yahoo!により提案されている [13]．DomainKeys では，送信側は DNS に公開鍵を保存しておき，MTA がメールの内容を秘密鍵で署名し，それをヘッダを埋め込んで送信する．受信側では公開鍵を取ってきて署名が本物であるかどうかを確認することができる．

4. 送信元認証におけるハンドシェーク活用の提案

前節の SPF や Sender ID は，ラストワンホップの SMTP クライアントの IP アドレスを認証している．そのため，転送されるメールや，ML などから配られるメールの大元の送信元アドレスを認証することができない．また，電子署名を使った手法が近い将来に広く実現されることも想定し難い．そこで本稿では送信側のサーバが送ったメールの情報を保持しておき，受信側のメールサーバは送信側のメールサーバに対し，メールを送ったことを問い合わせて確認してから受信するメール送信システムを提案する．

4.1 概　　要

予めメールサーバの管理者はそのドメインのユーザに問い合わせに応答するサーバの名前を知らせておくこととする．そして，ユーザが送信元を信頼してほしいような責任のあるメールを送る場合は，

`X-HSRS: <サーバ名>`

というヘッダをつけてメールを送ることとする．

もしそのヘッダがついていた場合は，MTA は 1 つ 1 つのメールに固有な ID を作成し，

`X-HSID: <ID>`

というヘッダをつけて宛て先 MTA に転送する．

それを受信した側は送信元に問い合わせて確認が取れたら

`X-HSResult: Success`

というヘッダをつけてユーザのメールボックスに配信する．

4.2 送信元アドレスの定義

本においては，送信元アドレスとはエンベロープではなく，最終的にユーザの MUA に表示されるヘッダの “From:” フィールドと定義する．すなわち，“From:” フィールドのドメイン名をメールの送信に責任のあるドメインとし，そのドメインの管理する DNS に問い合わせるサーバの IP アドレスを載せておく．

4.3 問い合わせサーバの割り出し

受信側が ID を問い合わせるサーバ名は「“X-HSRS:” フィールドのホスト名」,「.(ピリオド)」,「“From:” フィールドのドメイン名」をつなげたものとする．例えばヘッダ情報が

`X-HSRS: mail-a`
`From: alice@example.com`

であった場合には，mail-a.example.com を DNS 検索する．

これにより，データベースを共有していれば，メールの送信とデータベースへの問い合わせの応答を別のサーバにすることができる．また example.com の送信サーバが東京 (192.168.1.1) と大阪 (192.168.1.2) に 2 台あり，別々のデータベースを使用している場合や，他のドメインである example.net の送信サーバ (172.20.1.1) を使わせてもらう場合には，DNS に以下のように記述しておけばよい．

```
reply-tokyo.example.com. A 192.168.1.1
reply-osaka.example.com. A 192.168.1.2
reply-net.example.com. A 172.20.1.1
```

メールの管理者は，東京のサーバを使うユーザには “reply-tokyo”，大阪のサーバを使うユーザには “reply-osaka” を “X-HSRS:” フィールドの中身としてを使ってくださいというように指示しておく必要がある．

4.4 メール固有の ID の求め方

送信側の MTA は，送るメール 1 つ 1 つに固有の ID をつけ，それを後からの問い合わせに答えるためにデータベースに保存する．その ID とは，From，To，Subject，Date のフィールドをつなげたものを MD5 によりハッシュをかけたものとする．

`<ID> = MD5( From || To || Subject || Date)`

これは悪意のあるユーザが，送信されるメールの内容を途中で覗き見し，SPAM メールにそれと同じ ID を付けて，他の人に送信しようとすることを防ぐためである．ハッシュ関数の性質から，受信側の MTA や MUA で ID を再計算することで各フィールドが偽られていないことを確認できる．

4.5 ID の問い合わせ方法

メールの受信側から送信側に向けての ID の問い合わせは，拡張した SMTP コマンドを使用し，SMTP セッションとして行う．具体的には，以下のコマンドを打つことによりその ID がリストに存在するか問い合わせる．

`XRPY <ID>`

問い合わせに答える SMTP サーバは XRPY コマンドを実装し，これが実行された場合はデータベースを検索し，ID がリストに存在すれば応答コード 250 を返す．

4.6 結果の表示

受信側サーバは問い合わせの結果，ID がリストに存在することが確認できたら

`X-HSResult: Success`

というヘッダを，仮に失敗した場合は

`X-HSResult: Error`

をつけて，ユーザのメールボックスに配信する．後者のメールは SPAM メールである可能性が高い．

4.7 ID リストの消去

送ったメールが複数に転送される可能性があるため，問い合わせがあった ID をすぐにデータベースのリストから消去することはできない．ID は，作成から 7 日間たったものはリストから消去するようにする．

4.8 全体の処理の流れ

alice@example.com が bob@example.net に向けて，自分のドメインの MTA(mail-a.example.com) から相手先 MTA(mail-b.example.net) を経由してメールを送信する場合を例に，Fig.2 を用いて説明する．尚，Fig.2 においてはメールの送信サーバと問い合わせに答えるサーバは同一のものとしてある．

1　まず，alice の MUA は mail-a に接続をし，“X-HSRS: mail-a” というヘッダをつけたメールを bob 宛てに送信する．

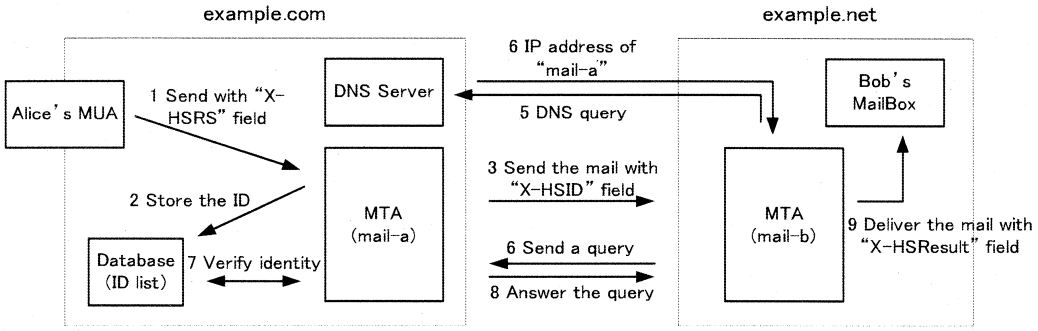

Fig. 2 Message Flow in Handshake Mail System.

2 mail-a は，受け取ったメールに "X-HSRS:" フィールドが存在した場合，ID を作成しデータベースに保存する．

3 その後 mail-a は，"X-HSID: <ID>" ヘッダを頭につけ，宛て先メールアドレスから mail-b を探し出し，SMTP を用いて mail-b にメールを送る．

4,5 mail-b は，受け取ったメールに "X-HSID:" フィールドが存在した場合，ID を問い合わせるサーバ名を割り出し (詳細は後述)，DNS を検索する．

6 mail-b は，5 で得た IP アドレスに接続し，ID がリストにあるかを尋ねる．

7,8 問い合わせのあったサーバは，データベースを検索し，結果を返す．

9 mail-b は，問い合わせの結果を "X-HSResult:" フィールドとして頭につけて，ユーザのメールボックスに配信する．

4.9 MTA への実装

本システムを，MTA は qmail を，データベースは PostgreSQL を用いて実装した．1 つの MTA が，送信サーバと受信サーバの両方を役割を果たしている場合は，「受け取ったメールヘッダに，X-HSRS だけ存在したら X-HSID をつけて送信 (転送)，X-HSID も存在したら ID を送信側に問い合わせて受信」という処理が必要である．処理の流れ図を Fig.3 に示す．これらは，メールメッセージを配送キューに格納するプログラムである qmail-queue にラッパをかぶせることで実現した．その結果，正しく認証された場合のメッセージは Fig.4 のようになり，仕様どおりに動くことが確認された．

5. 考　　察

本システムは，ドメインレベルでの送信元認証と位置づけることができる．そのため電子署名と比較した利点は，ユーザ側からみると，MUA のソフトウェアを変える必要がなく，従来と同様のメールの送受信において，

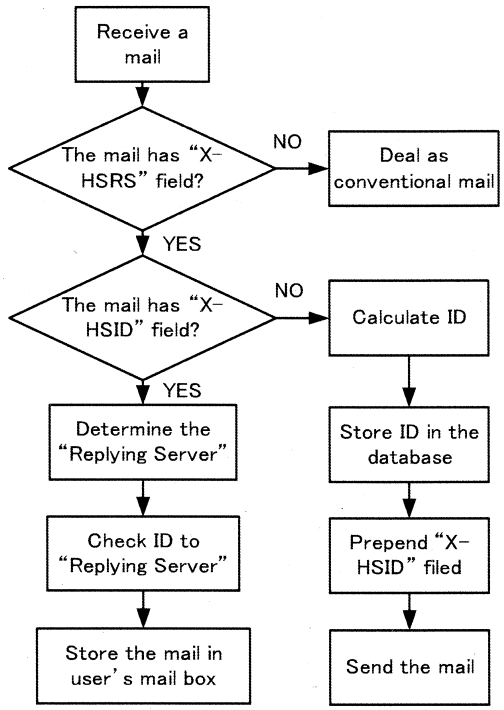

Fig. 3 Process Flowchart of MTA.

MTA で送信元認証が行われているという点である．ただし，MUA によっては "X-HSRS" ヘッダをつけて送れないものも存在する．本システムでは，ユーザがメールに特殊なヘッダをつけた時のみ，送信元認証の仕組みが働くようにしたが，送信元を偽装し SPAM メールを完全に食い止めるためには，サーバ側で全てのメールに ID をつけ，全てのメールの問い合わせに答えるようにしなければならないと考えている．

送信側 MTA と受信側 MTA の片方，もしくは両方が本システムに対応していない場合は，従来どおりのメー

```
Return-Path: <alice@example.com>
Delivered-To: bob@example.net
Received: (qmail 4800 invoked from network); 6 Oct 2004 15:37:33 +0900
X-HSResult: Success
Received: from mail-a.example.com (172.20.1.1)
 by mail-b.example.net with SMTP; 6 Oct 2004 15:37:22 +0900
Received: (qmail 10973 invoked from network); 6 Oct 2004 15:36:35 +0900
X-HSID: b7bd84ee13d89f05dee30e5f9ec471e9
Received: from AlicePC.example.com (192.168.1.1)
 by 172.20.1.1 with SMTP; 6 Oct 2004 15:36:20 +0900
X-HSRS: mail-a
Date: Wen, 6 Oct 2004 15:35:55 +0900
To: bob@example.net
From: alice@example.com
Subject: Test mail

This is a test mail.
```

Fig. 4 An Example of Email Message after successful verification.

ルとして処理される．すなわち，送信側 MTA が本方式に対応していない場合はヘッダに何も記載されていないため，従来どおりのメールとして受信され，受信側 MTA が本方式に対応していない場合も，“X-HS***” フィールドは理解不能なヘッダなため無視されてそのまま受信される．

本システムが SPF や Sender ID に対して優れている点は，転送されてきたメールや，ML から送られてきたメールも一番大元の送信サーバに対して確認することが可能であるという点である．これは SPF や Sender ID において，SPAM 業者が SPAM メールをあたかも他から転送されて来たかのごとく見せかけて送った場合，受信側で送信元認証を行うことができないことに比べ，優れている点であると言える．

また，DomainKeys のような公開鍵暗号を使う手法に比べると，鍵の管理などの手間を省くことができる．

本システムは，従来の SMTP に比べて，ヘッダを解析しなければならないことと，問い合わせのコネクションを張ることから，処理にかかるオーバーヘッドが増加するという欠点がある．また悪意のあるユーザからの攻撃としては，送信元を偽ったメールをたくさん投げることによる問い合わせサーバに対する DoS 攻撃が考えられる．そして，正規のメールの送信を覗き見することで，ID を構成する要素を全て真似たメールの再送攻撃を行うことも可能である．

6. まとめ

本稿では送信元認証の 1 方式として，送信側のサーバが送ったメールの情報を保持しておき，受信側のメールサーバは送信側のメールサーバに対し，メールを送ったことを問い合わせて確認してから受信するメール送信システムを提案した．まだプロトタイプの実装段階のため，今後は再度仕様を吟味し，実装していく予定である．

現在のメールは，一対一，もしくは一対多 (ML など) に送る機能しか果たしていないため，このシステムを応用したメールのより高度な機能の実現も目指していきたい．本システムにおいて送信サーバで保持している ID を別の用途に活用することによって，メールでのアンケートに対する複数の返信メールを自動的に集計したり，多人数による協調作業を支援したりするシステムの構築について検討する予定である．

参考文献

[1] Spam Statistics, Brightmail. http://www.brightmail.com/spamstats.html

[2] Deborah Fallows, “How It Is Huring Email and Degrading Life on the Internet,” http://www.pewinternet.org/

[3] OECD Work on Spam, http://www.oecd.org/sti/spam

[4] Eric Allman, “Features: Spam, Spam, Spam, Spam, Spam, the FTC, and Spam,” ACM Queue vol. 1, No. 6, Sep. 2003.

[5] Anti-Spam Research Group (ASRG), http://asrg.sp.am/

[6] Internet Mail Consortium ietf-mxcomp, http://www.imc.org/ietf-mxcomp/index.html

[7] The White House, http://www.whitehouse.gov/

[8] “迷惑メール関係施策,” http://www.soumu.go.jp/index.html

[9] Mail Abuse Prevention System, http://mail-abuse.org/

[10] Open Relay DataBase, http://ordb.org/

[11] Sender Policy Framework, http://spf.pobox.com/

[12] M. Wong and M. Lentczner, “The SPF Record Format and Sender-ID Protocol,” IETF Internet Draft draft-ietf-marid-protocol-03.txt, Aug. 2004.

[13] Mark Delany, “Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys),” IETF Internet Draft draft-delany-domainkeys-base-01.txt, Aug. 2004.