ディスプレイネームをユーザ認証に利用したなりすましメール送信防止システムの試作

王 建人[1]　Natthamon Pongchanchai[2]　山井 成良[1]　北川 直哉[1]　Vasaka Visoottiviseth[2]

概要：差出人を詐称した「なりすましメール」は標的型攻撃の発端として頻繁に利用され，その対策は急務である．特に正規ユーザのアカウントを不正入手し，ディスプレイネーム（表示名）を詐称したなりすましメールを発信されると，従来の送信ドメイン認証技術を適用してもディスプレイネームの詐称を検出する効果は期待できない．そこで本稿では，[1] で提案されている普段使用するディスプレイネームを予め登録しておき，送信メッセージ中のディスプレイネームが MSA に登録されたものと一致しない限りメッセージの送出を許可しないようにする手法を採用し，システムとして実装する．本システムメール送信時間に与える影響も実験により測定した．これにより，たとえアカウント情報が漏洩したとしてもなりすましメールの送信を抑制する効果が期待できる．

キーワード：電子メール，なりすましメール，サブミッションスパム，ディスプレイネーム

A Prototype of Spoofed Email Submission Prevention System Using Display Name as a User Authenticator

Kento Ou[1]　Natthamon Pongchanchai[2]　Nariyoshi Yamai[1]　Naoya Kitagawa[1]
Vasaka Visoottiviseth[2]

Keywords: email, spoofed email, submission spam, display name

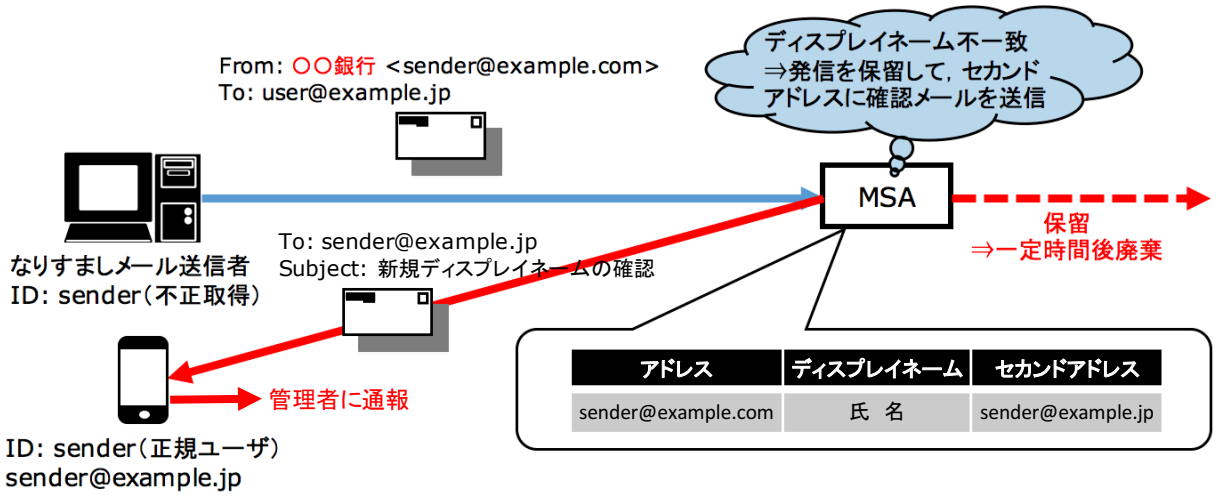
1. はじめに

電子メールは個人や企業など広く使われており，社会活動を支える重要なコミュニケーション手段の 1 つである．一方，電子メールは不特定多数のユーザとメッセージをやり取りできることからセキュリティ上多くの問題を抱えており，特に広告，フィッシング詐欺，マルウェア配布などを目的に不特定多数のアドレス宛に一方的に送りつけられる迷惑メールの蔓延は大きな社会問題となっている．最近では大規模ボットネットの摘発，OP25B（Outbound Port 25 Blocking）[2] など，迷惑メール送信を抑止する取り組みが進みつつあり，電子メール全体に占める迷惑メールの割合は 2010 年夏頃の約 90%から最近では約 55%まで減少してきている [3], [4]．しかし，迷惑メール送信の手口も巧妙化されてきており，たとえば OP25B を回避するためにプロバイダ等が運用する MSA（Message Submission Agent）の正規の ID とパスワード（以下，これらをまとめてアカウント情報と呼ぶ）を不正な方法で取得し，これらを用いて正規ユーザになりすまして MSA から迷惑メールを送信する手口[*1]が横行している [5]．

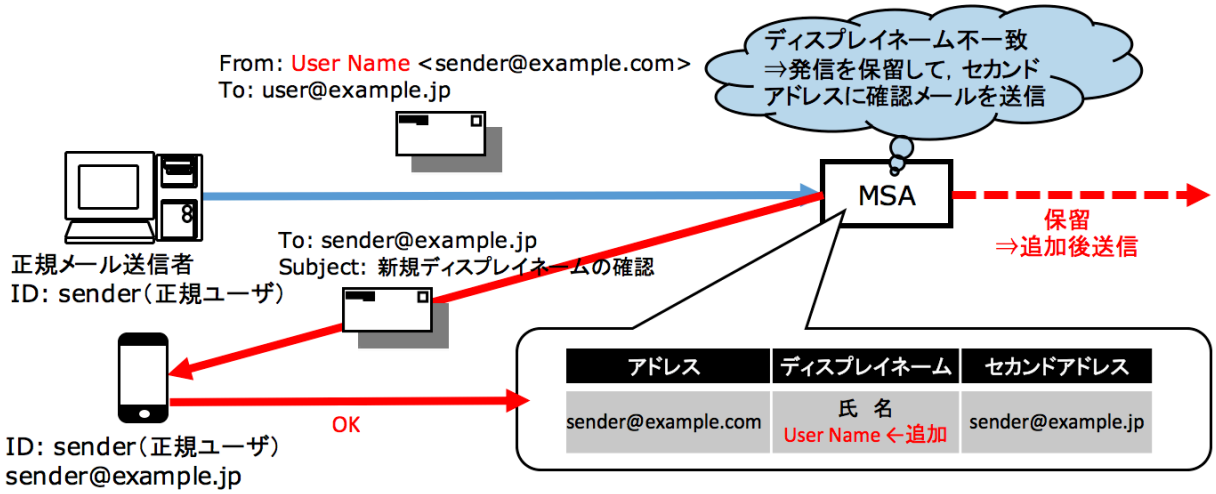
また，多くの迷惑メールでは，差出人のメールアドレスや，差出人や宛先の氏名を表記するために用いられるディスプレイネーム（表示名）を詐称した「なりすましメール」となっており，このようなメールを受信したユーザが騙されるケースが後を絶たない．特に，標的型攻撃では，2015

[1] 東京農工大学
Tokyo University of Agriculture and Technology
[2] Mahidol University, Thailand

[*1] 一般に submission spam，MSA 踏み台送信などと呼ばれる．



(a) なりすましメール送信者が送信を試みた場合



(b) 正規ユーザが送信を試みた場合

図 1 ディスプレイネーム不一致時の動作例

年 5 月に発生した日本年金機構の事例 [6] や，2016 年 3 月に発生した JTB 子会社の事例 [7] などにおいて，攻撃の発端としてなりすましメールが用いられ，被害が深刻になってきている．

このような迷惑メール，特に差出人アドレスを詐称したなりすましメールへの対策として，SPF[8]，DKIM[9]，DMARC[10] 等の送信ドメイン認証技術が知られている [2]．現時点では普及率がそれほど高くないなどの問題点はあるものの，これらの技術を用いれば，差出人アドレスの詐称についてはこれを正しく検出できる効果がある程度期待できる．ところが，差出人や宛先の氏名を表記するために用いられるディスプレイネーム（表示名）は利用可能な文字列にあまり制限がなく，送信者が RFC5322.From（ヘッダ From）[11] に付随して設定された文字列がそのまま表示されることが多い．そのため，差出人アドレスは詐称せずに正規ユーザのものをそのまま使用し，ディスプレイネームだけ実在の企業や人物に詐称したなりすましメールを正規の MSA 経由で発信されると，従来の送信ドメイン認証技術ではこれを検出できず，被害がより深刻化する可能性がある．

この問題に対して，文献 [1] では，ディスプレイネームをユーザ認証に利用する手法が提案されている．この手法により，アカウント情報が漏洩した場合でもディスプレイネームを詐称したなりすましメール送信を防止する効果が期待できる．文献 [1] では，手法の提案にとどまり実装方法については述べられていない．そこで本稿では，この手法に基づいたシステムを試作し，その実装方法について述べる．第 2 章で手法の概要を述べた後，第 3 章でシステム構成や実装環境などについて述べ，第 4 章で実装したシステムがメール送信に与える影響を測定した結果を示し，第

5 章にてまとめる．

2. 既存の踏み台送信対策手法

文献 [5] では既存の踏み台送信対策として SMTP 認証と送信通数制限を行う方法と，送信者詐称の制限を行う方法が示されている．本章ではこれらの対策の概要と問題点を述べる．

2.1 SMTP 認証と送信通数制限

この方法では，SMTP 認証を用いて送信者 ID を特定し，送信者 ID からの単位時間あたりの送信数を制限する．送信サーバにて単位時間あたりの送信数が閾値を超えた場合，SMTP 機能を自動で停止する．その後管理者がユーザへ確認などの対応を行う．

この方法によって単位時間あたりに多くのなりすましメールを送信するホストは特定できるが，例えば，30 分に 1 回なりすましメールを送信するといったように単位時間あたりのなりすましメール送信数が少ない場合には効果がない．さらに悪意のない一般ユーザであっても，送信数によって制限される可能性がある．

2.2 差出人メールアドレス詐称の制限

この方法では，MSA において送信者認証を行った後，メッセージの差出人アドレスなどの送信者情報が認証されたユーザに対応付けられたものかどうかチェックを行うことによりなりすましを見分ける．このチェックに失敗した場合には送信を許可しないものである．そうすることで差出人メールアドレスを詐称した MSA 踏み台送信を抑止する効果が期待できる．

しかしこの方法では差出人メールアドレスを詐称は防止できるが，ディスプレイネームだけを詐称したなりすましメールの送信は防止できない．

3. 実装する方式の概要

本章では文献 [1] で提案されている手法の概要を述べる．前章で述べたように，送信ドメイン認証技術では，特に差出人メールアドレスは詐称せず，ディスプレイネームだけを詐称したなりすましメールを MSA が踏み台にして送信された場合，これを防止することはできない．

ここで，ディスプレイネームの通常の使い方を考察すると，正規ユーザはたとえ複数の端末を利用する場合であっても個々の端末上の MUA にユーザ名，パスワードなどとともにディスプレイネームを登録しており，これらのディスプレイネームを変更することは稀である．また，MSA 踏み台送信を行う不正送信者は通常は乗っ取ったアカウントの正しいディスプレイネームを知っていない．また，複数の MUA を使用している場合では，MUA 毎の設定により常に同一のディスプレイネームが使用されるとは限らない．これらの特徴に着目すると，ディスプレイネームは複数の正答を持つ一種のパスワードとして利用することが可能である．

そこで，本システムでは予め MSA に送信者が使用するディスプレイネームをデータベースに登録しておき，それが RFC5322.From に付随するディスプレイネームと一致しない限りメッセージの送出を許可しないようにする．MSA に登録するディスプレイネームは複数でもよく，そのいずれかに RFC5322.From に付随するディスプレイネームが一致すれば送信を許可するようにする．送信が中止されたメールはユーザには返送されず，MSA によって保留される．

さらに，ディスプレイネーム不一致時にメールが保留された場合，データベースに登録されているユーザのセカンドアドレス宛に認証メールが送信される．認証メールからユーザが認証することで，使用可能なディスプレイネームとしてデータベースへ登録され，保留されていたメールが送信される．この認証メールによるメリットは 2 点挙げられる．1 つめのメリットは，悪意のあるユーザのアカウント不正利用を即座に知ることが可能な事である．**図 1**(a) のように，アカウント情報を不正取得したユーザがなりすましメールを送信した場合，ユーザへの認証メールによって不正利用を知ることができ，すぐに管理者に通報し対策を講じることが可能である．2 つめのメリットは，自身でディスプレイネームを変更した場合，事前に登録する作業がなくなるため，ユーザの負担軽減が期待できる事である．その場合のシステムの動作例を図 1(b) に示す．

本システムは，ディスプレイネームを認証に用いるため，2.1 節にて示した単位時間あたりのなりすましメール送信数が少ないホストに対しても有効であると言える．さらに 2.2 節にて示したディスプレイネームのみを詐称したなりすましメールについても同様に有効であるといえる．なお，本システムでは事前に使用するディスプレイネームを登録する作業または認証メールによるディスプレイネーム認証作業が必要になるため，従来のメールシステムよりユーザの負担が増加する恐れがある．しかし，正規ユーザはディスプレイネームを変更することは稀であるため，登録作業を行う必要回数が少なく，ユーザ負担は許容できる範囲であると考えられる．

最後に保留メールについて述べる．メール保留中に，その保留メールと同じディスプレイネームが使用された場合，一時拒否応答を返す．すなわち，1 種類のディスプレイネームにつき保留できるメールは 1 通までである．ただし，本システムを実際に運用する際の問題点として，ディスプレイネームを送信毎に変更してなりすましメールが送信されることにより保留メールが膨大な量になり，システムの動作に影響を与える可能性が挙げられる．それを防止するために，保留メールの有効期限を設定する．定期的に

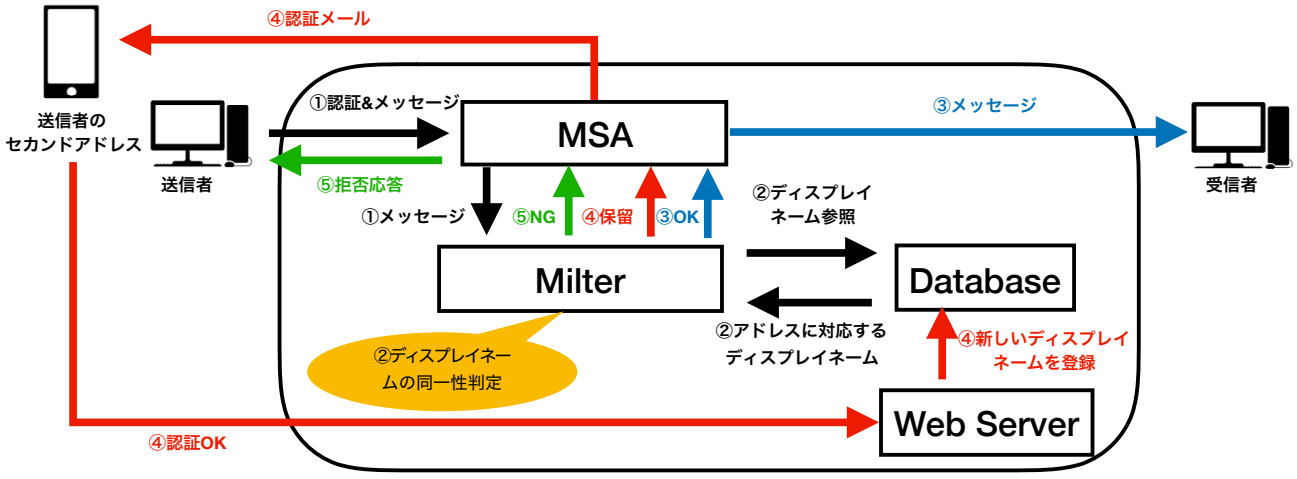


図 2 提案システムの構成と動作

保留メールを監視し，有効期限が超過しているものがあれば削除する．

4. システムの実装

4.1 システムの構成

今回実装するシステムの構成を**図 2** に示す．また，本システムの想定している動作を以下に示す．なお，各ステップの番号は図 2 中の番号に対応している．

(1) MSA は端末との間でユーザ認証を行い，これに成功すると端末からメッセージを受け取る．そして MSA は milter にメッセージ（ヘッダ及び本文）を送る．

(2) milter プログラムは MSA からユーザ ID とメッセージを受け取ると，ヘッダから RFC5322.From およびこれに付随するディスプレイネームを取り出し，アドレスをキーとしてデータベースから得たディスプレイネームリストと照合を行う．

(3) メッセージ中の RFC5322.From がデータベース中のアドレスと一致し，さらにメッセージ中のディスプレイネームがデータベース中のディスプレイネームリストのいずれかの要素と一致すれば，milter はこのメッセージを受理する応答を MSA に返す．MSA は端末に受理応答を返し，このメッセージを送信する．

(4) メッセージ中のディスプレイネームが一致せず，かつそのディスプレイネームが初めて使われた場合，milter はメッセージを保留する応答を MSA に返す．MSA はメッセージを保留キューに隔離し，送信者のセカンドアドレスへ認証メールを送信する．認証が完了するとデータベースへ新たにディスプレイネームが登録される．その後，保留キューに保留されていたメッセージが送信される．

(5) メッセージ中のディスプレイネームが一致せず，かつ

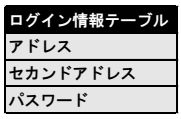


図 3 ログイン情報テーブル

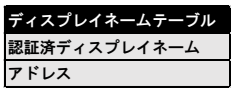


図 4 ディスプレイネームテーブル

そのディスプレイネームを使用したメールがすでに保留されている場合，保留処理は行わず，送信元へ拒否応答を返す．この際一時的なエラーを意味する SMTP ステータスコード 400 番台を返す．

4.2 実装環境

システム実装において使用した環境を述べる．まず，MSA にはメールサーバソフトウェアである postfix[12] を導入した．milter プログラムは python 言語のライブラリである pymilter[13] を導入し python 言語で作成した．Web インタフェース (Web サーバ) は Apache[14] を導入し，PHP を用いて作成した．保留メールの有効期限をチェックするスクリプトは python 言語で作成した．

4.3 データベース設計

本システムで用いるデータベースの設計について述べる．作成したテーブルは以下の 4 種類であり，それぞれのテーブルにおいて必要な要素と，どのような場合に参照されるかを述べる．

- ログイン情報 (ユーザ情報) テーブル (**図 3**)
- ディスプレイネームテーブル (**図 4**)
- 未認証ディスプレイネームテーブル (**図 5**)

- キュー ID テーブル (**図 6**)

4.3.1 ログイン情報 (ユーザ情報) テーブル

Web インタフェースへのログイン情報とセカンドアドレスを登録するテーブルであり，アドレス，セカンドアドレス，パスワードの 3 つ組レコードである．パスワードは安全のためハッシュ化して登録する．また，メールのアカウント情報が漏洩した場合でも，Web インタフェースへのログインを不可能にするため，パスワードはメールのアカウント情報とは異なるようにする．本システムでは，すでに異なるパスワードがログイン情報テーブルに登録されているものとする．Web インタフェースへのログイン時，ディスプレイネーム不一致時に参照される．

4.3.2 ディスプレイネームテーブル

ユーザが使用可能なディスプレイネーム (以下，認証済ディスプレイネームと呼ぶ) を登録するテーブルであり，認証済ディスプレイネーム，アドレスの 2 つ組レコードである．認証済ディスプレイネームは複数登録可能であり，milter プログラム内でのディスプレイネーム一致判定の際，アドレスをキーとして認証ディスプレイネームをすべて参照し，ユーザが使用したディスプレイネームと一致するものが存在するか照合する．さらに，このテーブルは Web インタフェース内でのディスプレイネーム登録・削除時，認証後のディスプレイネーム登録時に参照される．

4.3.3 未認証ディスプレイネームテーブル

未認証のディスプレイネームを登録するテーブルであり，アドレス，未認証ディスプレイネーム，ワンタイムトークンの 3 つ組のレコードである．milter プログラム内でのディスプレイネーム一致判定が不一致であった場合，使用されたディスプレイネームが未認証ディスプレイネームとして，認証用のワンタイムトークンと共に登録される．認証完了後，本テーブルからディスプレイネームテーブルへ移動する際に参照される．移動後，本テーブルから該当レコードは削除される．未認証ディスプレイネームは複数登録可能であるが，等価なものは登録不可である．

4.3.4 キュー ID テーブル

保留メールのキュー ID を登録するテーブルであり，キュー ID テーブル，アドレス，未認証ディスプレイネーム，時刻の 4 つ組のレコードである．キュー ID とは postfix がメール送信時にメールごとに割り当てる ID である．本テーブルでは，保留メールを識別可能にするために，キュー ID 取得後に登録される．ディスプレイネームの認証完了後，アドレスと未認証ディスプレイネームをキーとしてキュー ID を参照し，それに対応するメールを再送する．また，保留期間のチェックのために時刻が定期的に参照される．認証完了が確認された場合や，有効期限の超過が確認された場合，該当レコードは削除される．

未認証ディスプレイネーム一時保存テーブル
アドレス
未認証ディスプレイネーム
ワンタイムトークン

図 5 未認証ディスプレイネームテーブル

キューIDテーブル
キューID
アドレス
未認証ディスプレイネーム
時刻

図 6 キュー ID テーブル

Verification for your E-mail and display name
2017年7月10日 15:37

Someone try to send message from your email sender@example.jp with user1 display name. Please verify by clicking the following URL link:
http://disp.net.cs.tuat.ac.jp/auth.php?urltoken=a3c35b78-8ca9-41b1-b748-a60543327c0b

図 7 認証メール

4.4 milter プログラムの実装

milter プログラムでは，まず MSA から受信したメッセージのアドレスとディスプレイネームを文字列処理によって取りだす．次にディスプレイネームテーブルから，アドレスをキーとしてディスプレイネームのリストを取り出し，メッセージのディスプレイネームと一致判定を行う．こちらの判定処理は，空白などを含む完全一致で行っている．一致する場合，処理をせずにメッセージを送出するために，milter で「続行」を意味する “continue” の戻り値 (pymilter では Milter.CONTINUE) を返す．

ディスプレイネームが不一致の場合，初めにアドレスと未認証ディスプレイネームがすでに未認証ディスプレイネームテーブルに登録されているかを確認する．すでに同じ未認証ディスプレイネームが存在した場合，メールを拒否するため，milter で「拒否」を意味する “reject” の戻り値 (pymilter では Milter.REJECT) を返す．同じ未認証ディスプレイネームが存在しない場合，メッセージを保留キューに隔離して 2 要素認証に移行する．保留キューへの移動は，milter で「隔離」を意味する関数を実行することで行われる．pymilter では self.quarantine() である．この関数は，milter プログラムのメール処理において，メッセージ全体の処理が終了した時にのみ実行可能なため注意が必

要である．

また，認証メールによって認証が完了後，メッセージの送信を再開する必要があるため，保留されたメールを識別可能にする必要がある．そのために，MSA がメッセージそれぞれに割り当てるキュー ID と呼ばれる ID を取得し，新しく使用されたディスプレイネームと共にキュー ID テーブルへ格納する．milter でのキュー ID の取得方法は sendmail macro の “i” を用いることで参照できる (pymilter では self.getsymval('i'))．認証には有効期限を設けるために，保留キューに移動した時刻もキュー ID テーブルへ格納している．

最後に認証メールについて述べる．milter プログラムはメッセージ保留後，認証メールを送信する．認証は Web サーバで行うため，認証を行う PHP ファイルの URL にワンタイムトークンを加えたものを認証用 URL とし，認証用メールに記す．ワンタイムトークンは認証を完了するためのキーとして使用する．重複を避けるため python の UUID モジュールの uuid4 関数を用いて生成する．生成されたワンタイムトークンは 16 バイトの 16 進数のランダムな文字列で構成されている．認証メール作成後，ワンタイムトークンを含めた 3 つ組のレコードを未認証ディスプレイネームテーブルへ格納する．ユーザが認証メール受信後，認証用 URL をクリックするとワンタイムトークンが Web サーバに渡され，それに対応するアドレスと未認証ディスプレイネームを取得する．その 2 つをキーとしてキュー ID テーブルからキュー ID を取得し，保留メールの再送を行う．また，未認証ディスプレイネームはディスプレイネームテーブルに登録し，今後使用可能にする．以上でディスプレイネームの認証処理は完了となる．認証後，関連するテーブルのレコードは削除する．**図 7** に認証メールの例を示す．この例では “sender@example.jp” のアドレスで未認証ディスプレイネーム “user1” の使用が確認されたことをユーザへ提示し，認証するためには画面下部の URL をクリックすることを促している．URL 内の “urltoken=” 以降の文字列はワンタイムトークンである．

以上のように認証メールにはアドレスと使用された未認証ディスプレイネームの情報が記載されている．これについて，例えばメール本文の内容をすべて記載するといったように，更に多くの送信メールの情報をユーザに提示することで，メールアカウントが悪用されたかどうかの判断が容易になることも考えられる．しかしなりすましメールはディスプレイネームを詐称するという特徴から，本稿では使用された未認証ディスプレイネームの記載のみで十分であると判断し，そのように実装した．

4.5 Web インタフェースの実装

Web インタフェースで実装する機能について述べる．

4.5.1 ユーザのログイン処理

ユーザが使用するディスプレイネームを事前に登録可能にするために，ログイン処理を実装した．ログイン後，ユーザは自由にディスプレイネームの登録と削除を行うことが可能となる．ログイン後，アドレスに対応するディスプレイネームリストをディスプレイネームテーブルから取得し，**図 8** に示すような管理画面に，アドレスに対応するディスプレイネームリストをすべて表示する．

4.5.2 ディスプレイネームの登録・削除処理

ディスプレイネームの登録と削除を行う処理を実装した．図 8 の Add display name の下のテキストボックスに，追加したいディスプレイネームを記入し，Add ボタンを押すことで，ディスプレイネームテーブルに新たなディスプレイネームが登録され，それ以降メール送信に使用することが可能になる．ディスプレイネームの削除についても同様である．

4.5.3 2 要素認証の認証処理

ユーザが認証メールの URL をクリックした時，認証を承認する処理を実装した．URL からワンタイムトークンを受け取り，それをキーとしてキュー ID テーブルを参照する．対応するキュー ID と未認証ディスプレイネーム取得後，まずディスプレイネームテーブルに未認証ディスプレイネームを認証済ディスプレイネームとして登録する．その後，キュー ID に対応するメッセージの送出コマンドを MSA に送信する．送出コマンドは “postsuper -r キュー ID” である．すべての処理が完了後，該当レコードを削除

図 8 Web インタフェース

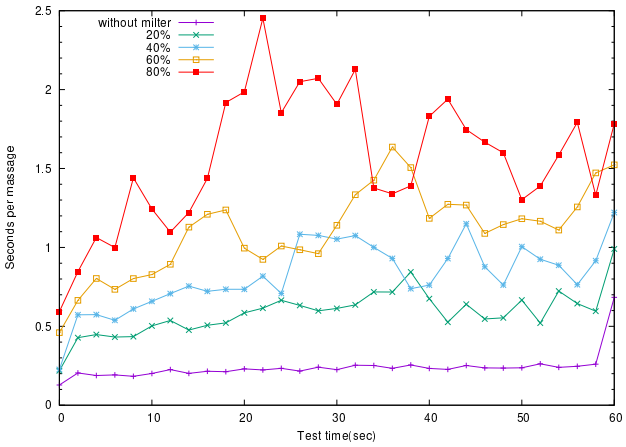


図 9 測定結果

表 1 1 メールあたりの送信時間の平均

ディスプレイネーム詐称メールの割合	Milter 適用なし	20%	40%	60%	80%
1 メールあたりの送信時間の平均 (sec)	0.207	0.591	0.845	1.149	1.624

する．

4.6 保留メールの有効期限監視スクリプトの実装

保留メールの有効期限を監視するためのスクリプトの実装について述べる．スクリプトの流れを順に記述する．初めに，キュー ID テーブルからすべてのレコードを参照し，キューへ移動した時刻を取得する．次に現在時刻を取得し，キューへ移動した時刻との差を求める．その差が設定した有効期限を過ぎていた場合，保留キューからキュー ID に対応するメールを削除し，該当するレコードをキュー ID テーブルから削除する．保留キューからメールを削除するコマンドは “postsuper -d キュー ID” である．有効期限は自由に設定することが可能である．本システムでは試作システムのため保留メール量が少ないことから，有効期限を 24 時間に設定したが，実際の利用の場合は保留メール量に対して適切な有効期限の検討が必要であると考えられる．スクリプトを定期的に実行するために，Linux の cron 機能を利用した．本システムでは 5 分毎に実行されるように設定した．

5. 評価実験

評価実験として，本システムの導入がメール送信スピードに与える影響について測定する．測定には mstone[15] と呼ばれるベンチマークソフトを用いて行う．mstone は SMTP，POP，IMAP など様々なプロトコルについてテストが可能なソフトウェアである．さらに，テストに用いるメールの指定や，メールヘッダの書き換えが可能なため，ディスプレイネーム詐称メールの送信テストを行うことが可能である．

5.1 実験内容

MSA に 100 個のユーザを登録し，そのユーザが同時に 1 分間，メール送信を繰り返した場合の，メール 1 通あたりの送信時間を測定する．初めに milter プログラムを適用せずに測定した．次に，milter プログラムを適用し，テストメール全体の 20%をディスプレイネーム詐称メールとして測定を行った．その後ディスプレイネーム詐称メールの割合を 40%，60%，80%と増加させ同様に測定を行った．ただし，mstone ではメール 1 通ごとのディスプレイネームの変更ができないため，評価実験を行っている間は未認証ディスプレイネームテーブルのレコードの重複を許可するように設定した．

5.2 結果と考察

メール 1 通あたりの送信時間の測定結果を**図 9** に示す．milter プログラム適用なしの場合，送信時間は 0.2[秒/通] 付近を保ち安定している．ディスプレイネーム詐称メールの割合を増加させた場合，データベースへのアクセスや認証メールの送信処理も同時に増加するため，結果として送信時間が増加している．さらに，速度も安定しておらず，80%の場合，ピーク時 2.5[秒/通] を記録した．また，それぞれの場合についての送信時間の平均を**表 1** に示す．milter 適用なしの場合と 80%の場合を比較すると，80%の送信時間が milter 適用なしの送信時間の約 7.8 倍となった．

以上の結果より，本システムの導入によるメール送信時間への影響は大きいと言えるが，ディスプレイネームの一致判定やデータベースの参照は問題なく動作していたことから，本システムの動作への致命的な影響は無いと考えられる．しかし，今後本システムが ISP などで採用された場合，メール送信量が更に増加すると考えられるため，プログラムの高速化を図ることは必要であると言える．

6. おわりに

本稿では，ディスプレイネームが一種のパスワードとして機能することに着目し，ディスプレイネームをユーザ認証に利用したなりすましメール対策システムを実装した．本システムでは，未認証ディスプレイネームの使用のために 2 要素認証を必要とするため，アカウントの不正使用を即座に検知可能である事を示した．負荷テストでは，本システムがメール送信時間へ与える影響を示し，プログラムの高速化が必要であることを述べた．また，本システムの有効性を評価するためには，多くのユーザが登録されており，かつ MSA 踏み台送信の被害が確認された状況下で評価する必要があり，大規模な社会実験を行う必要がある．そのため，本稿執筆時までには有効性を評価することは困

難であったが，今後評価が必要であると言える．

謝辞 本研究の一部は科学技術振興機構平成 29 年度地域産学バリュープログラム (課題番号 VP29117941338) の補助を受けている．ここに記して感謝の意を表する．

参考文献

[1] 山井成良：“ディスプレイネームをユーザ認証に利用したなりすましメール対策手法”，マルチメディア，分散，協調とモバイル (DICOMO 2017) シンポジウム論文集，8D-4，pp.1665–1670，情報処理学会，2017 年 6 月

[2] 迷惑メール対策推進協議会：迷惑メール対策ハンドブック 2016 (オンライン)，入手先 ⟨http://www.dekyo.or.jp/soudan/image/anti_spam/book/2016/2016MHB_all.pdf⟩ (参照 2017-05-04)，2016 年 12 月．

[3] Symantec Corporation: State of Spam & Phishing Report — A Monthly Report, No.45, Symantec Corporation (online), available from ⟨https://www.symantec.com/content/dam/symantec/docs/security-center/archives/spam-report-sept-10-en.pdf⟩ (accessed 2017-05-04).

[4] Symantec Corporation: Internet Security Threat Report, Vol.21, Symantec Corporation (online), available from ⟨https://www.symantec.com/content/dam/symantec/docs/security-center/archives/istr-16-april-volume-21-en.pdf⟩ (accessed 2017-05-04), April 2016.

[5] 迷惑メール対策推進協議会 技術ワーキンググループ: 電気通信事業者による迷惑メールの踏み台送信対策の状況 (概要)(オンライン)，入手先 ⟨https://www.dekyo.or.jp/soudan/data/anti_spam/20161124fumidai.pdf⟩ (参照 2017-10-30)，2016 年 12 月．

[6] サイバーセキュリティ戦略本部: 日本年金機構における個人情報流出事案に関する原因究明調査結果，内閣サイバーセキュリティセンター (オンライン)，入手先 ⟨http://www.nisc.go.jp/active/kihon/pdf/incident_report.pdf⟩ (参照 2017-05-04)，2015 年 8 月．

[7] 株式会社ジェイティービー: 不正アクセスによる個人情報流出の可能性について —現状報告と再発防止策—，株式会社ジェイティービー (オンライン)，入手先 ⟨https://www.jtbcorp.jp/jp/160824.html⟩ (参照 2017-05-04)，2016 年 8 月．

[8] Kitterman, S.: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1, RFC7208, IETF, April 2014.

[9] Crocker, D., Hansen, T. and Kucherawy, M.: DomainKeys Identified Mail (DKIM) Signatures, RFC6376, IETF, September 2011.

[10] Kucherawy, M. and Zwicky, E (Eds.): Domain-based Message Authentication, Reporting, and Conformance (DMARC), RFC7489, IETF, March 2015.

[11] Resnick, P. Ed.: Internet Message Format, RFC5322, IETF, October 2008.

[12] The Postfix Home Page (online), available from ⟨http://www.postfix.org/start.html⟩ (accessed 2017-09-04)

[13] Stuart D. Gathman,Business Management Systems Inc.: Sendmail/Postfix Milters in Python (online), available from ⟨https://pythonhosted.org/milter/⟩ (accessed 2017-09-04)

[14] The Apache Software Foundation: Apache HTTP SERVER PROJECT available from ⟨https://httpd.apache.org⟩ (accessed 2017-09-04)

[15] SourceForge.net:The Mstone multi-protocol benchmarking system (online), available from ⟨http://mstone.sourceforge.net/doc-4.9/mstone.html⟩ (accessed 2017-09-04)