L-028

メールユーザエージェントによる送信ドメイン認証の実験

An Experiment on the Sender Domain Authentication by a Mail User Agent

山田 真也† 野口 健一郎†
Shinya Yamada Kenichiro Noguchi

1. はじめに

迷惑メールなどはメールアドレスを偽装して送られる場合が多い．メールでのなりすましを防ぐ対策としてメール受信側の MUA(メールユーザエージェント，すなわちメールクライアント)で送信ドメイン認証をする実験をした．

送信ドメイン認証はメール受信側の MTA（メール転送エージェント，すなわちメールサーバ）で検証することが一般的である．しかし，メール送信側が送信ドメイン認証に対応していても，受信側の MTA も対応していなければ認証はできない．そこで，受信側で MTA が認証に対応していなくても，MUA が送信ドメイン認証を行うことを可能にした．

送信ドメイン認証には，IP アドレスと電子署名を用いるものがある．本実験では，前者には SPF[1]，後者には DKIM[2]という認証方式をベースとして MUA が同時になりすましを検証できるメールクライアントを試作した．

2. 背景

送信ドメイン認証とは，送信者のメールアドレスが正しいものであるかを確かめる技術である．そのメールのドメインをもとに，それが正しい送信者から送信されているか否かを検証する．

SPF (Sender Policy Framework) は，送信されたメールが送信側ドメインで許可された IP アドレスから送信されたものかどうかを受信側 MTA が送信側の DNS を通じて検証する．

DKIM (DomainKeys Identified Mail) は，送信側 MTA が DKIM-Signature ヘッダに署名を追加し，また受信側 MTA が送信側の DNS を通じて発信者の公開鍵を調べることで署名の正当性を検証する．

DKIM の送信側の MUA ライブラリはあるものの[3]，受信側の MUA で認証を行うものは依然としてないのが現状である．

3. MUA による SPF の実験

(1) 認証の流れ

MUA による SPF 認証の流れは次の通りである(図 1).

①送信側の MUA は送信側の MTA にメールを送信.
②送信側の MTA は受信側の MTA を含むメールサーバへ転送.
③受信側の MUA はメールサーバからメール受信を要求.
④受信側の MUA はメールサーバからメールを受信.
⑤受信側の MUA は受信したメールのヘッダから IP アドレスを調べる.
⑥受信側 MUA は送信側 DNS へ SPF レコードを問合せる.
⑦送信側 DNS は受信側 MUA に SPF レコードを返す.
⑧受信側 MUA は SPF レコードから IP アドレスを検証.

†神奈川大学大学院理学研究科情報科学専攻

SPF レコードとは送信側ドメインでメール送信の許可，または不許可をしている IP アドレスを記述したものである．

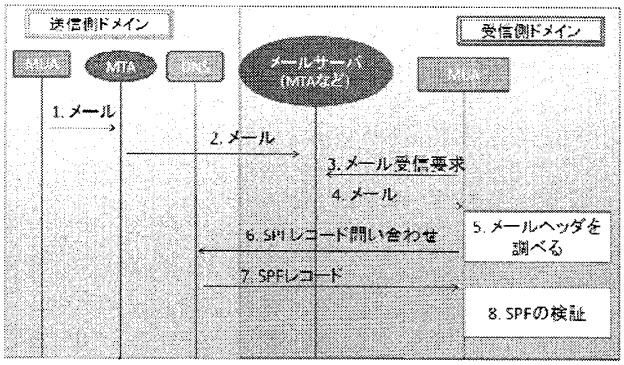

図 1. MUA による SPF のシーケンス図

(2) MUA による検証方式

受信したメールのヘッダから，送信された IP アドレスを Received ヘッダをもとに調べる．送信者（Return-path, From, Sender ヘッダのいずれか）のドメインの DNS に SPF レコードを問い合わせる．その記述の中に調べた IP アドレスがあれば承認する．

4. MUA による DKIM の実験

(1) 認証の流れ

MUA による DKIM 認証の流れは次の通りである(図 2).

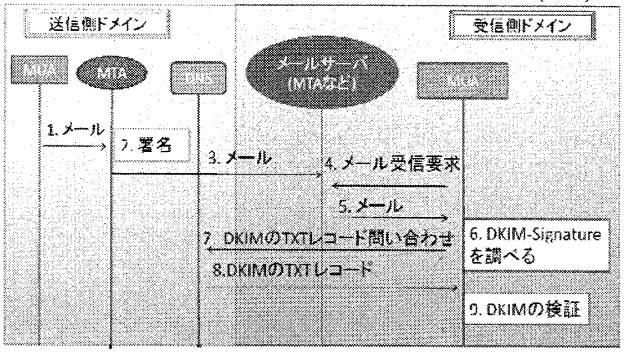

図 2. MUA 版 DKIM のシーケンス図

①送信側の MUA は送信側の MTA にメールを送信.
②送信側の MTA はメールに署名データを含む DKIM-Signature を付ける.
③送信側の MTA は受信側の MTA を含むメールサーバへ転送.
④受信側の MUA はメールサーバからメール受信を要求.
⑤受信側の MUA はメールサーバからメールを受信.
⑥受信側の MUA は受信したメールのヘッダから DKIM-Signature を調べる.
⑦受信側 MUA は送信側 DNS へ DKIM の TXT レコードを問合せる.

⑧送信側 DNS は受信側 MUA に DKIM の TXT レコードを返す.
⑨DKIM の検証を行う.

DKIM-Signature には主に次のタグが設定される．図 3 に例を示す.

- a タグ：署名のアルゴリズム.
- s タグ：セレクタ．公開鍵を取得するときの DNS クエリを構成する要素となる.
- d タグ：署名を行ったドメイン名.
- c タグ：ヘッダとボディの正規化方法の指定．ハッシュを取る前に正規化することで認証しやすくする. simple と relaxed がある.
- h タグ：署名対象のヘッダリスト.
- bh タグ：ボディ部のハッシュデータ.
- b タグ：ヘッダと DKIM-Signature の b タグの値を除いた部分と秘密鍵で作成した署名データ.

```
DKIM-Signature: v=1; a=rsa-sha256;
c=relaxed/simple;
 d=sendmail.net; s=gatsby; t=1264630679;
  bh=ZKrsHyOWxeU/oy8wUwOiYVKO9DhS2pNTHOuxCCd5jSg=;
  h=Date:Message-Id:Content-Type:Content-Transfer-
Encoding:
   MIME-Version:From:To:Subject;
  b=qYAGiKP2FQIZUm/sd5vVOc4PxdIe/yyVwsgcHmAdeq9AT1U
  cUHJd/6wDwW+P7GpnFOINeICxjIm4thJcYQHMQ8xwbMDuoTEn
  VoQjSEi5DNnVwIfK/oyWkU9sdmbjVfK2fXu
   UDcy2psZOU2hq2HpaqajpVYdL/vXIpdUgO/1IprU=
```

図 3．DKIM-Signatrue の例

(2) MUA による検証方式

受信したメールの DKIM-Signature ヘッダから s タグと d タグを用いてクエリを作成し，MTA が送信したドメインの DNS に DKIM の TXT レコードを問合せることで公開鍵を取得する．また，メールのボディ部とヘッダを c タグにしたがって正規化する．メールのボディ部のハッシュを取り，DKIM-Signature ヘッダの bh タグと一致するか調べる．最後にヘッダと DKIM-Signature(署名部分除く)のハッシュを取り，公開鍵と b タグの署名から検証する．ヘッダとボディとの結果が正当なら承認する.

5. 実装

言語は Java で実装し，メールの扱いについては Java メールの API を用いて実現した．具体的には MUA は JavaMail のサンプルファイルを改良し，受信用 MUA に機能追加して送信ドメイン認証ができるようにした.

検証部分に関しては DNS の問い合わせを外部プロセスとして実行して，結果を読み込んだ．署名の検証は Java の暗号の API を用いて実現した.

図 4 に示すように，メールの内容を表示したときに送信ドメイン認証のボタンを押すと検証結果が表示されるようにした.

1 つの MUA で SPF と DKIM の両方の認証が同時に行えるようにした．検証結果が同時に表示されるので，ユーザがそれぞれの認証が合格しているかを確認できるようにした.

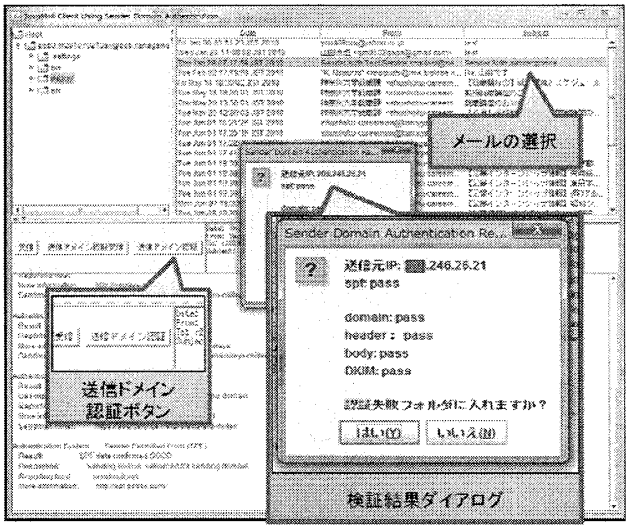

図 4．認証結果をダイアログで表示した MUA

6. 考察

DKIM を導入した MTA はまだ少ないが，本実験により MUA が個々に対応することで検証が可能になった.

SPF は本来，送信者の情報を SMTP エンベロープの「MAIL FROM」を調べるべきであるが，本実験では MUA で実装したため，ヘッダの送信者情報を利用した．これにより，実際の送信者である「MAIL FROM」と異なる場合が生じ，偽装される恐れがあるが，MTA がメールを転送する際に記述される Received ヘッダ情報の妥当性のチェックをすることにより偽装を調べることができると考えられる.

DKIM は SMTP のエンベロープ情報を使用しないので MUA でも問題なく実装できた.

7. 終わりに

MUA でも送信ドメイン認証を実現できた．これによりユーザの判断でそのなりすましメールの処理を決めることができるようになった．例えば，MTA が検証に失敗したメールの受信を拒否するように設定されているとすると，送信ドメイン認証に対応していない MTA から送られたメールをユーザは受信できないことになる．これを MUA が検証を行うことによって，検証結果に加えてユーザの目でメールの内容を確認し，そのメールの最終的な処理を自身で決定することができるようになった.

今後の課題は次の通りである.

(1) SPF における Received ヘッダでの送信者情報の妥当性チェック
(2) ドメインの評価情報を用いた更なる発展した検証
(3) SPF を発展させた Sender ID の導入

参考文献

[1] RFC4408: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail , Version 1
[2] RFC4871: DomainKeys Identified Mail (DKIM) Signature
[3]DKIM for Java AGITOS
http://www.agitos.de/dkim-for-java-mail-open-source-library-2.html