L-010

メールユーザエージェントによる送信者認証の実験: DKIM 方式の適用

An Experiment on the Sender Authentication by a Mail User Agent Applying DKIM

山田 真也† 野口 健一郎†
Shinya Yamada Kenichiro Noguchi

1. はじめに

電子メールのなりすましを防ぐための技術である送信者認証の一つに DKIM（DomainKeys Identified Mail）がある。昨年度の FIT では受信側の MUA（Mail User Agent、すなわちメールクライアント）による送信ドメイン認証の実験を発表した[1]。本発表では、送信側 MUA による DKIM への対応を含めて、DKIM を用いたユーザの送信者認証の実験を行った結果を報告する。

2. 背景

2.1 送信ドメイン認証

送信ドメイン認証とは、受信者が受け取ったメールについて、送信者情報が詐称されていないかどうかをドメインごとに確認する技術である。

DKIM (DomainKeys Identified Mail) [2][3]は，送信者がメールに署名情報を含む DKIM-Signature ヘッダを追加・送信し、受信者が送信側の DNS を通じて送信者の公開鍵を調べて署名検証を行うことで送信者の正当性を検証する．他に IP アドレスベースの SPF（Sender Policy Framework）がある。

一般に DKIM は MTA（Mail Transfer Agent、すなわちメールサーバ）で実装される場合が多く、送信側および受信側の MUA で署名・認証を行うものは少ないのが現状である。

2.2 送信者認証

多くの MUA に実装される規格として PGP（Pretty Good Privacy） と S/MIME （ Secure/Multipurpose Internet Mail Extensions）があり、電子メールの暗号化と電子署名ができる規格である。署名は送信側が署名データを MIME 形式で添付して送信し、受信側が公開鍵を用いて検証を行う。S/MIME では認証局の公開鍵証明書を、PGP では別のユーザが署名した公開鍵を取得して検証する。

送信ドメイン認証を送信者認証と言うこともあるが、ここでは区別して考える。

3. MUA による DKIM の実験

3.1 概要

(1) 実験の目的

DKIM の RFC[2]では MUA による DKIM の実装は許されているが、実際は少ない。そこでこれを実験し、利点、課題を明確にする。

(2) 全体構成

図 1 に示す。

(3) 送信側 MUA の機能

(a) MTA で行う場合と同等のもの

- DKIM 署名付きメールの送信

(b) MUA で行うことで必要と考えられるもの

- MUA（ユーザ）毎の鍵ペアとレコードの作成
  各 MUA で個別の鍵が利用できるようにする。
- 署名するメールアドレスを限定した署名とレコードの作成
  RFC ではオプションではあるが対応させた。

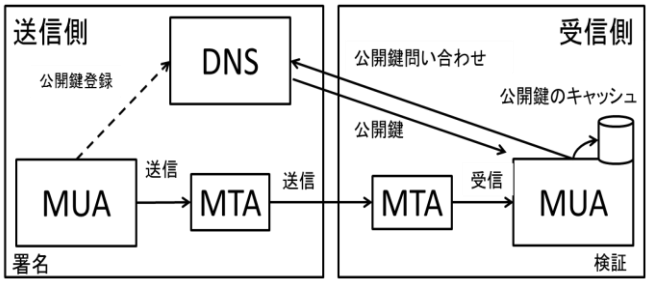

図 1. 全体構成

(4) 受信側 MUA の機能

(a) MTA で行う場合と同等のもの

- 送信側 DNS サーバへの公開鍵問い合わせおよび DKIM の検証

(b) MUA で行うことで必要と考えられるもの

- 受信したメールの認証結果の表示
  受信側 MTA の実装では結果をヘッダに挿入する。MUA では明示的に表示させることができる。
- 公開鍵のキャッシュ機能
  送信者が不特定多数の受信側 MTA と異なり、MUA ではやりとりをする送信者が限定できると考えられ、有効であると考えられる。

3.2 処理の流れ

(1) 鍵の作成と登録

① 送信側 DNS サーバ管理者に公開鍵レコードの識別子となるセレクタを決めてもらう。
② 送信側 MUA は秘密鍵と公開鍵のペアを作成する。
③ 送信側 MUA は公開鍵のレコードを作成する
④ 送信側のユーザは DNS サーバ管理者にレコードを登録してもらう。

(2) MUA によるメールの署名および送信

① 送信側 MUA はメールを作成する。
② メールの署名を作成する。
③ 署名データを含む DKIM-Signature を付ける。
④ 送信側の MUA は送信側 MTA へメールを送信する。
⑤ 送信側の MTA は受信側のメールサーバへ転送する。

(3) MUA によるメールの受信および検証

① 受信側の MUA は受信側のメールサーバからメールを受信する。
② メールのヘッダから DKIM-Signature を調べる.

†神奈川大学大学院理学研究科情報科学専攻

③　ハッシュマップにすでに送信側の公開鍵があればそれを用いる。
④　③で公開鍵がなければ受信側 MUA は送信側 DNS へ DKIM の TXT レコードを問い合わせて取得し、キャッシュする。
⑤　取得した公開鍵を使って DKIM の検証を行う.

3.3　DKIM の署名方式

MTA の場合と同様に署名は次の通りに行う。
①　ボディを正規化しハッシュをとる。
②　DKIM-Signature にボディハッシュ(bh タグ)を追加する。
③　署名対象のヘッダを抽出し、正規化する。
④　送信側の秘密鍵を使って署名データを作成する。例は図 2 のようになる。

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple;
d=nol.info.kanagawa-u.ac.jp;
    i=nlabuser@nol.info.kanagawa-u.ac.jp; s=select1;
h=Date:From:To:Subject;
   bh=xdsgrybRfaa2IlddkUlZ5VEMpsryvCnkIaGX5TH0tS4=;
   b=TiVjHeWk3Kon4PdIA4cWrzg7EpqYs8zaVTH8B9rgFan
YO06n22CtxlfVmFVL8fqNowIdqWfrXE5tuksUhz9Ss0dCi/RX
0rYtuzWvSz6n94azWDMu8hUO13fND3YEUhAM5SBdrLP+
vHWwMdaWEbBxE3VkKxTZNPqhLaFzRzp+XAM=
```

図 2.　DKIM-Signatrue の例

検証については次の通りに行う。
①　受信したメールの DKIM-Signature ヘッダからクエリを作成し、送信側 DNS サーバに公開鍵を問合せて取得、キャッシュする。
②　公開鍵を用いて検証を行う。
③　i タグが設定されている時はメールアドレスのユーザ名が指定されていることになり、DKIM TXT レコードの g タグの内容と一致するかを確認する。

4.　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。

実験ではコマンドラインから引数を指定して実行するようにした。メールの認証を行ったときに図 3 のように認証結果がダイアログで表示されるようにした。

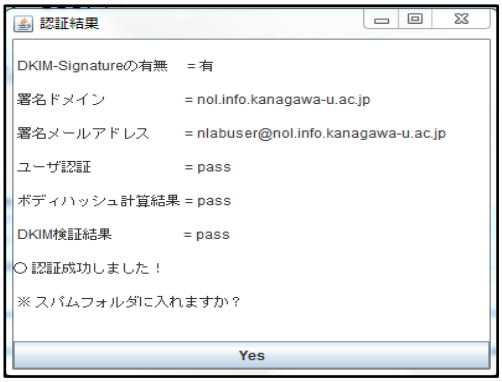

図 3.　認証結果をダイアログで表示した MUA

公開鍵はクエリをキーとして公開鍵、ユーザ名、登録年月日をハッシュマップに格納するようにした。また、その内容は図 4 のように出力できるようにした。登録年月日を設定したことにより一定期間経過した公開鍵は破棄または再取得することができるようにした。

```
1 クエリ= gamma._domainkey.example.com; 公開鍵=
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIhyR3oItOy22ZOaBrIVe9m/iME
3RqOJeasANSpg2YTHTYV+Xtp4xwf5gTjCmHQEMOs0qYu0FYiNQPQogJ2t0Mfx9zNu
06rfRBDjiIU9tpx2T+NGlWZ8qhbiLo5By8apJavLyqTLavyPSrvsx0B3YzC63T4Ag
e2CDqZYA+OwSMWQIDAQAB; ユーザ名= *; 登録年月日= 2011-7-1
2 クエリ= nlabuser20110701._domainkey.nol.info.kanagawa-u.ac.jp;
公開鍵=
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQChXoFKIW9vkcUDh9BKsOZ55Xszw
2G1+sXy+xH56jA6VeABq9ukkEa6C8X7hLjO7ir79z6lO6jx1DeggSlOiIwlA47by7
mNkGBLxQoYSq2EkKUKrBJZBP0XPEzrWBejR0T4DILWD3Dvk9iBylNZunq4MCLSne5
0kd9+1OuL60BdnwIDAQAB; ユーザ名= nlabuser; 登録年月日= 2011-7-1
```

図 4.　ハッシュマップの内容

5.　考察

(1)　互換性について

昨年の実験に続き、DKIM 認証には受信側 MUA は送信側 MTA、MUA からの署名に対応させた。送信側 MUA は受信側 MUA、MTA の検証に対応させ、互換性がある。

(2)　MUA と MTA による方式の違い

MTA に比べて MUA では署名できるヘッダが限定されてしまうが、鍵を MUA（ユーザのメールアドレス）毎にしたことでセキュリティが向上できると考えられる。

MUA のマシンで秘密鍵の管理方式は改善が必要である。

(3)　S/MIME などの送信者認証との違い

DKIM による送信者認証は受信側の公開鍵の取得に DNS サーバを用いている点が異なる。S/MIME のように認証局を使うこともなく、PGP のように鍵の更新が分かりにくい web of trust （信用の網）を用いない点で認証がしやすい。ユーザにとっての DKIM の更なる利便性の向上には DNS サーバに鍵レコードの登録を自動化することや、別の鍵の交換方式の提案が課題である。

(4)　キャッシュ機能について

MTA が DKIM の認証をするのであれば多数のユーザの公開鍵をキャッシュすることになる。しかし MUA がキャッシュするのであれば頻繁にやりとりをする送信者は限定される。したがって MUA が公開鍵をキャッシュすることは有効であると考えられる。

鍵データをキャッシュすることにおいては受信側 DNS キャッシュサーバも公開鍵データを含む DKIM TXT レコードをキャッシュしている可能性がある。しかし一般にレコードをキャッシュする有効期限である TTL(Time To Live)は数時間～3 日程度で設定されるため、レコードの保持時間としては短い。ユーザがどの程度公開鍵キャッシュの有効期限を設定することが適切であるかはこれからの検討課題である。

6.　今後の課題

(1) 送信側の公開鍵レコードの登録方法の検討
(2) 送信側における秘密鍵の管理方式の検討
(3) 公開鍵の他の取得方式の検討
(4) 公開鍵のキャッシュ有効期限の検討

参考文献

[1] 山田真也, 野口健一郎：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 , FIT2010
[2] DomainKeys Identified Mail (DKIM) Signature, RFC 4686, September 2006
[3] Analysis of Threats Motivating DomainKeys Identified Mail (DKIM), RFC4686, September 2006