DNSSEC/SMIMEA 対応 MUA アドオンの実装

M2013SC005 平林 有理
指導教員 河野 浩之

1 はじめに

近年，DNS キャッシュポイズニング攻撃が大手 ISP で観測されている [1]．これまでは，DNS スプーフィングを受けても，重要な通信では TLS を利用するため大事には至らないと考えられてきた．しかし，2011 年に発生した DigiNotar 事件を受け，この認識を改めなければならなくなった [4]．DigiNotar 事件を受けて注目されているのが，DNS-based Authentication of Named Entities (DANE)[2] である．しかし，Mail User Agent (MUA) には未だに対応していない．

本研究では，DANE による Secure Multipurpose Internet Mail Extension (S/MIME) 証明書検証を可能にする MUA 用のアドオン，SMIMEA Validator を作成する．SMIMEA Validator は，SMIMEA リソースレコード (SMIMEA RR) と呼ばれる証明書情報を，メール送信者が属するゾーンの DNS サーバから取得し，メール送信者の S/MIME 証明書を検証する機能を備える．SMIMEA RR の仕様については，インターネットドラフトである draft-ietf-dane-smime-07[3] に準じる．また，SMIMEA Validator は，DNSSEC/TLSA Validator[2] を基に実装し，マルチプラットフォームでの動作を目標とする．

検証実験では，レジストラからドメインを取得し，DNSSEC/DANE 対応の DNS サーバ (bind9 1:9.9.5.dfsg) に規定のリソースレコードを登録することで，実インターネットを用いた実験環境を構築した．評価では，GNU/Linux, Windows, Mac OSX, *BSD 向けに SMIMEA Validator をビルドし動作検証する．加えて，証明書検証にかかる時間の計測及び，本アドオンの有無による Thunderbird の起動時間を計測することで，パフォーマンスを評価する．

本研究論文は全 6 節から構成される．2 節では本研究の前提とするインターネット標準について解説する．3 節では提案手法について説明する．4 節では使用するプログラムの実装を詳説し，5 節ではシステムの評価を述べる．6 節ではまとめを示す．

2 本研究の前提とするインターネット標準

この節では，本研究の前提となる規格である DNSSEC, DANE, S/MIME の概要と類似の先行研究についてそれぞれ 2.1 節, 2.2 節, 2.3 節で述べる．

2.1 DNSSEC

DNS Security Extensions (DNSSEC) は DNS 応答にデジタル署名を付与することで，その出自と完全性を証明する DNS の拡張仕様である [1]．DNS スプーフィング対策として近年注目されている．

2.2 DANE

DANE は，DNSSEC によって信頼できる通信基盤となった DNS を利用して，証明書情報をユーザへ伝達し，ユーザが利用するサービスの証明書を検証する仕組みである [2]．DANE を利用することで証明書の「ピン留め」や，サービス管理者が自身で発行する自己署名証明書に信頼性をもたせることが可能となる．

2.3 S/MIME

S/MIME は，現在，広く用いられている電子メールの暗号化と，デジタル署名に関するインターネット標準である．これまでの利用形態では，公開鍵の安全性を Trusted Third Party (TTP) である認証局に依存しており，利用するには認証局に証明書を発行してもらう必要があった．しかしながら近年，TTP を認証局ではなく，DNSSEC/-DANE を有効にした DNS にすることで，認証局に依存しない S/MIME 証明書検証システムが提案された [3]．

3 設計

ここでは，本研究で実装する SMIMEA Validator の開発環境及び，設計について述べる．

3.1 SMIMEA Validator の開発環境

本研究で実装する SMIMEA Validator は，Web ブラウザ向けの DNSSEC/TLSA 検証アドオンである DNSSEC/TLSA Validator を元に作成する．開発 OS はビルド環境が整っているという理由から Ubuntu 14.04 LTS 64bit を利用した．また，最初の動作対象 MUA として，Mozilla Thunderbird を選定した．これは第 1 に，Thunderbird がマルチプラットフォームに対応している点．第 2 に，開発元が同一である Mozilla Firefox で，DNSSEC/TLSA Validator の動作が確認できた点．第 3 に，Firefox と Thunderbird には，アドオンのフォーマットに一定の互換性があり，DNSSEC/TLSA Validator の資源を再利用できる点．の 3 点の理由からである．

3.2 DNSSEC/TLSA Validator

DNSSEC/TLSA Validator は，CZ.NIC で Martin Straka らによって開発され，GPL v3 で公開されている Web ブラウザアドオンである．このアドオンを利用することで，Web ブラウザで表示するサイトのドメインを DNSSEC で検証でき，サーバ証明書を DANE で検証できる．DNSSEC, DANE の検証には ldns[3], OpenSSL[4] 及び，Unbound[5] を用いて，C 言語で作成された共有ライブラリを利用している．

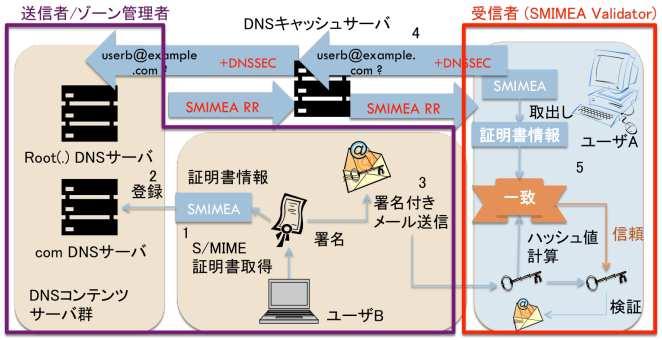
[1] http://jprs.jp/tech/security/2014-04-15-portrandomization.html
[2] https://www.dnssec-validator.cz
[3] http://www.nlnetlabs.nl/projects/ldns/
[4] https://www.openssl.org/
[5] http://unbound.net/index.html

3.3 動作概要

この節では SMIMEA Validator の動作概要について述べる．SMIMEA Validator は，MUA で受信した署名付きメールを表示した際に，対象メールアドレスの DNS サーバに対して SMIMEA RR の送信を要求し，それを元に S/MIME 証明書を検証をする．

図 1 は送信者及び受信者が予め設定する手順も含めたシステム全体の動作概要図である．

図 1 システム動作概要図



ユーザ B 及びドメイン管理者の作業は次のようになる．

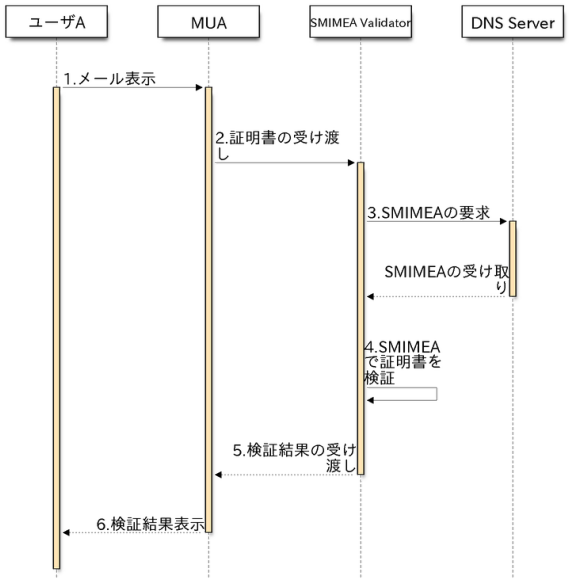
1. ユーザ B はユーザ B の S/MIME 証明書から SMIMEA を生成し，example.com のゾーン管理者に SMIMEA RR の登録を依頼する．
2. example.com のゾーン管理者はユーザ B の SMIMEA RR を登録する．
3. ユーザ B がユーザ A に S/MIME による署名付きメールを送信．ユーザ A が受け取る．
4. SMIMEA Validator によって，署名者のメールアドレスのドメイン (example.com) に対し，SMIMEA RR を要求．当該 DNS コンテンツサーバから取得する．
5. SMIMEA Validator は SMIMEA RR によって S/MIME 証明書を検証し，証明書の正当性を確認する．

また，図 2 は SMIMEA Validator をインストールした MUA で検証するユーザ側の動作に注目したシーケンス図である．

SMIMEA Validator は次のような手順で証明書を検証する．

1. ユーザ A は MUA 上で署名付きメールを表示する．
2. MUA は SMIMEA Validator へ S/MIME 証明書を受け渡す．
3. SMIMEA Validator は DANE によって SMIMEA を DNS サーバから取得する．
4. SMIMEA Validator は SMIMEA を利用して，MUA から受け取った S/MIME 証明書を検証する．
5. SMIMEA Validator は MUA に証明書の検証結果を返す．
6. MUA は結果を表示しユーザへ通知する．

図 2 SMIMEA Validator のシーケンス図



一連の動作によってユーザ A は，署名付きメールに利用された証明書が正当なものであることを検証できる．

4 実装

この節では，SMIMEA Validator の実装について述べる．

4.1 共有ライブラリの動作

SMIMEA Validator では，DNS サーバへの SMIMEA の要求/取得と，SMIMEA を利用した S/MIME 証明書の検証に共有ライブラリを用いる．DNSSEC/TLSA Validator には既に DNS コンテンツサーバへの問い合わせ機能，TLSA RR を利用した証明書の検証機能については実装されている．このため，本研究では，MUA から渡されたメール情報から SMIMEA の DNS クエリを生成する機能を実装した．また，共有ライブラリの動作を確認するために，共有ライブラリのソースコードを単体で動作可能なプログラムとしてビルドし，テストプログラムを作成した．以下に実行結果を示す．

```
1  DANE: Initialising DANE.
2  ===中略===
3  SMIMEA: Input parameters: domain='smimea.red';
       local_part='crttest'; label='smimecert'; options
       =3; resolver_address='(null)';
4  SMIMEA: Using system resolver.
5  DANE: crypto: SHA-224
6  SMIMEA: Domain is secured by DNSSEC ... found
       SMIMEA record(s).
7  ===中略===
8  SMIMEA: smimea.red: dnssec: SECURE (1), cert
       usage: 3, selector: 0, matching type: 0, assoc.
       hex: 3082052F30820417A0030201020...
9  ===中略===
10 DANE: result: 13
11 DANE: Main result: 13
12 DANE: Deinitialising DANE.
```

このテストでは，対象の DNS コンテンツサーバから SMIMEA RR を取得し，テストプログラム中に埋め込んだ証明書を検証した．指定した local_part，ドメイン名から SMIMEA RR を取得できていることが確認できる．また，検証のステータス 13 は，Certificate Usage 3 の SMIMEA で証明書の検証に成功したことを示している．

4.2 アドオンのパッケージ構成

Thunderbird のアドオン作成には，規定の形式に則ったパッケージングが必要となる．ここでは，本研究で作成，および変更したアドオンのファイルについて述べる．

install.rdf
: アドオンに関する情報が格納された XML ファイル，アドオンの名前や ID，バージョン，Thunderbird との互換性情報などを記述する．

chrome.manifest
: Thunderbird を拡張するリソースを定義するファイル．Thunderbird では，chrome と呼ばれる URI にアドオンのファイルをマッピングすることで利用する．

thunderbirdOverlay.xul
: Thunderbird の GUI に SMIMEA Vlidator のコンポーネントを追加するファイル．本研究では標準で Thunderbird に組み込まれている msgHdrViewOverlay.xul をオーバーレイする．msgHdrViewOverlay.xul は S/MIME の検証機能を Thunderbird に提供している．S/MIME の検証結果表示ボックスに DANE での検証結果を表示するアイコンを追加し，使用する Javascript ファイルを指定する．

smimea.js
: SMIMEA 検証の共有ライブラリと Thunderbird をつなぐ処理をする Javascript ファイル．Thunderbird から証明書を受け取り，smimealib.js に post する．

smimealib.js
: SMIMEA 検証の共有ライブラリを Thunderbird 上で非同期に利用するための Javascript ファイル．smimea.js から post されたドメイン情報，証明書情報などを js-ctypes を用いて共有ライブラリに渡す．また，その結果を smimea.js に post する．

libDANEcore-Linux-x86_64.so
: ネイティブで動作する DNSSEC 及び SMIMEA 検証の共有ライブラリ．

各ファイルの関係を図 3 に示す．

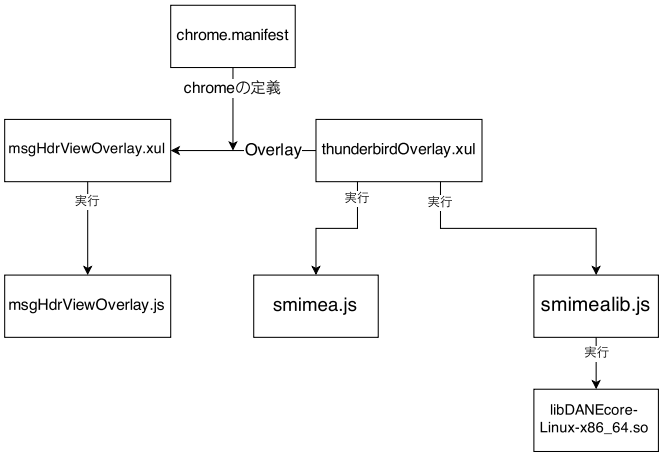
4.3 利用可能な検証

SMIMEA Validator を利用することで可能となる機能を表 1 に示す．SMIMEA Validator を利用することで，DANE での一連の SMIMEA 検証が可能となる．

表 1 SMIMEA Validator で可能となる機能

機能	Certificate Usage	対応
DANE 証明書検証	0, 1, 2, 3	✓
PKIX 証明書検証	1	✓(MUA に搭載)
暗号化/復号	-	✓(MUA に搭載)

図 3 ファイルの関係



5 評価

SMIMEA Validator を利用した証明書検証が，様々な環境で正常に動作することを確認する．また，SMIMEA Validator による証明書検証時間の測定及び，Thunderbird 起動遅延時間の測定をする．

5.1 DNS サーバの環境

アドオンが正常に動作することを確認するために，レジストラからドメインを取得し，DNSSEC/DANE に対応した DNS サーバを構築した．実際のインターネットを介して署名付きメールを検証することによって，SMIMEA Validator が有効に動作していることを確認する．本研究では，ドメインの DNSSEC 対応のために，ISC が提供している ISC's DNSSEC Look-aside Validation Registry[6] (ISC DLV Registry) を利用する．

表 2 に DNS サーバの環境を示す．

表 2 DNS サーバの環境

Model	HP ML110 G7
CPU	Intel®Celeron®Processor G530
OS	Ubuntu 14.04 LTS amd64
Memory	8GB
domain	smimea.red.
DNS Server	BIND9 (1:9.9.5.dfsg amd64)

5.2 他環境向けビルド

GNU/Linux, Windows, Mac OSX, *BSD 向けに SMIMEA Validator をビルドし，各 OS で動作する SMIMEA Validator を作成した．また，それぞれの OS にて SMIMEA Validator が正常に動作し利用できることを確認した．

[6] https://dlv.isc.org/

5.3 検証パフォーマンス計測

ここでは，SMIMEA Validator で証明書検証にかかる時間を測定する．テストはそれぞれ，表 3, 表 4 の 2 つのクライアント環境で実施した．

表 3 GNU/Linux 検証時間計測のクライアント環境

CPU	Intel®Core™i7-3930K Processor @ 4cores
OS	Ubuntu 14.04 LTS amd64 on VMware
Memory	8GB (DDR3-1600)
MUA	Thunderbird 31.3.0

表 4 Mac OSX 検証時間計測のクライアント環境

CPU	Intel®Core™i7-3540M CPU Processor
OS	Mac OSX 10.10.1
Memory	8GB (DDR3-1600)
MUA	Thunderbird 31.4.0

この実験では，Thunderbird 起動後に署名付きメールを選択してから，アドオンが証明書を検証し処理が完了するまでの時間を測定する．また，安定した計測結果を得るために，GNU/Linux と Mac OSX，2 つの環境で実験し，それぞれの項目について 20 回ずつ計測する．SMIMEA RR は Certificate Usage を 3, Slector を 0，Matching Type は 0, 1, 2 それぞれの時間を計測する．また，これに加えて，キャッシュの影響を計測するため，Matching Type 0 での検証後に，同一ドメインの別メールアドレス (Matching Type 0) にかかる時間を計測する．結果は，表 5 のようになった．

表 5 検証時間測定の結果

	GNU/Linux	Mac OSX
Matching Type 0	1424.25 msec	1399.9 msec
Matching Type 1	1236.15 msec	1239.1 msec
Matching Type 2	1381.45 msec	1425.05 msec
Matching Type 0 to 0	41.6 msec	52.65 msec

いずれの環境，Matching Type であっても 1.2〜1.4 秒程度の時間がかかった．検証済みドメインの，他メールアドレス検証にかかった時間は，50msec 程度であり，キャッシュの効果が大きいことがわかる．本研究では，DNS コンテンツサーバの DNSSEC 対応のために，国外にサーバを持つ ISC DLV Registry を利用しているため，DLV の問い合わせに時間がかかっている．このため，上位ゾーンに DS レコードを登録する運用形態の場合，パフォーマンス向上する可能性がある．

5.4 起動パフォーマンス計測

SMIMEA Validator のインストールによって，Thunderbird の起動時間に影響を与えるか測定する．測定には，Firefox 及び，Thunderbird の起動時間の計測が可能なアドオン，About Startup[7] を利用する．測定は，SMIMEA Validator を有効にした場合と無効にした場合で，それぞれ 20 回ずつ起動時間を測定しその平均値を比較することで行う．測定環境は，表 6 であり，計測結果は，表 7 の様になった．

表 6 起動時間測定の環境

CPU	Intel®Core™i7-3930K Processor @ 4cores
OS	Ubuntu 14.04 LTS amd64 on VMware
Memory	8GB (DDR3-1600)
MUA	Thunderbird 31.2.0

表 7 起動時間測定の結果

	平均時間	最大時間	最小時間
SMIMEA Validator 有効	1998.4 msec	2081 msec	1913 msec
SMIMEA Validator 無効	1998.7 msec	2083 msec	1903 msec

有意な差は計測できなかった．このことから，SMIMEA Validator が Thunderbird の起動時間に与える影響は，無視できる程度のものであると判断できる．

6 おわりに

本研究では，DANE による SMIMEA の検証を可能にする MUA 用のアドオン，SMIMEA Validator を実装した．また，アドオンが SMIMEA RR によってメールを検証に要する時間および，MUA の動作に与える影響を計測し評価した．評価では，SMIMEA の検証時間も 1.2〜1.4 秒程度となり，実用に耐える性能を示すことができた．加えて，MUA の起動時間への影響を計測し，ほぼ影響を与えないことを示した．

参考文献

[1] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: DNS Security Introduction and Requirements, *RFC 4033* (2005).

[2] Hoffman, P. and Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, *RFC 6698* (2012).

[3] Schlyter, J. and Hoffman, P.: Using Secure DNS to Associate Certificates with Domain Names For S/MIME, *Internet-Draft: draft-ietf-dane-smime-07* (2014).

[4] 中島智広：PKI の事故から学ぶ DNSSEC の必要性〜DNS スプーフィング攻撃の考察〜，DDNSSEC2013 スプリングフォーラム，http://dnsops.jp/event/20130529/dnssec2013springforum-nakashima-1.pdf (2013). Accessed 14/01/2015.

[7] https://addons.mozilla.org/ja/thunderbird/addon/about-startup/