MTA-STS : SMTP MTA Strict Transport Security による電子メール通信経路の暗号化とその実装

尾崎 周也[1,a)] 中島 博敬[2]

概要：電子メールは広く, コミュニケーションツールとして使用されている. SMTP は電子メール配送に用いられる標準的なプロトコルである. SMTP には配送するメッセージの秘匿性を担保するため, 複数の技術が用いられている. 代表的な手法として SMTP を拡張し, 暗号化された経路を用いる STARTTLS 拡張が存在するが, 日和見暗号であり, 中間者攻撃に脆弱である. 本論文では, JavaScript を用いて現在 IETF UTA Working Group で審議中の SMTP MTA Transport Security(MTA-STS) を実装し, サーバー間経路の暗号化を実現した. 評価では実装した MTA を用い, MTA-STS の可用性を検証した. また大手 ISP における MTA-STS への対応を調査した. その結果, MTA-STS への対応が不完全な ISP の存在や, MTA-STS に対応している ISP のすべてが検証モードで運用されていることを確認した.

Implementation of Encrypted Email Transport Using MTA-STS

SHUYA OSAKI[1,a)] HIROTAKA NAKAJIMA[2]

1. はじめに

電子メールはその登場以来, コミュニケーションツールとして支配的地位を占めている. 電子メールはビジネスの局面だけでなく, SNS によるコミュニケーションが主流になった現在に至っても私たちの生活に根ざし, これを支えている. しかしながらその普遍性と互換性の確保のためにしばしば, セキュリティが犠牲になっていると指摘される [1].

電子メールは Internet Message Access Protocol (IMAP), Post Office Protocol (POP) や Simple Mail Transfer Protocol(SMTP) などにより構成されている. SMTP は送信者と受信者のサーバーの間でメールを配送するプロトコルであり, 電子メールにおいて重要な役割を担っている.

SMTP は 30 年以上前に策定されたプロトコルであるため, メール配送サーバー (MTA) 間・サーバークライアント間の通信経路の秘匿化といったエンドツーエンドで電子メールを安全に配送する仕組みは備わっていない. 現在のインターネット環境は SMTP 設計時に想像されていたより脅威に溢れており, 時代の流れに伴い SMTP においてもプロトコルの安全性を向上させる仕組みが様々提案されてきた. 一方こうした仕組みは相互互換性を最大限に確保するため, 強制することはできない. その結果, 電子メールの安全性の底上げは限定的であった.

Zakir らによると, SMTP に備わる既存のセキュリティ技術の多くは大手メールサービスで施されているものの, 調査を行なった 700,000 を超える SMTP サーバーにおいて適切に運用されているのは 35%であった. STARTTLS は配送経路を暗号化ことにより, 中間者による傍受や改ざんを防ぐ SMTP の拡張であるが, 通信の開始時点で経路の秘匿化が期待できない, 日和見暗号であり, 中間者攻撃に脆弱であると指摘されている [2].

そこで本研究では, STARTTLS 拡張における中間者攻撃への脆弱性など, 安全な電子メール配送の不完全性について着目し, 独立した配送だけでなく一連の配送の暗号化に責任をもつ SMTP Strict Transport Security(MTA-STS)[3] を用いたセキュアな電子メール配送経路の確立と, 現状の調査をした.

[1] 慶應義塾大学 総合政策学部
Keio University Faculty of Policy Management
[2] 慶應義塾大学大学院 政策・メディア研究科
Keio University Graduate School of Media and Governance
[a)] shuya@sfc.wide.ad.jp

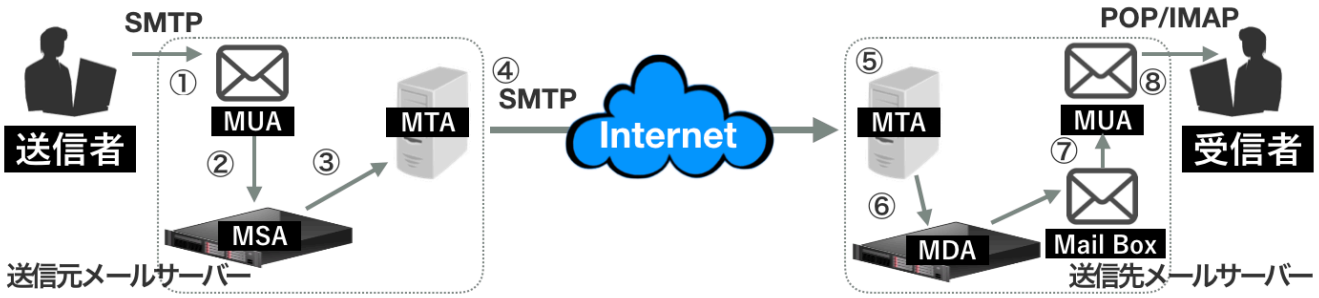

図 1 電子メール配送のエンドーツーエンドアーキテクチャとその通信プロトコル — 送信者は電子メールとして宛先と本文を作成する①. MUA が電子メールを中継の許可されたメールサーバーに転送する②. MSA が中継されたメールを認証し, Mail Transfer Agnet に中継する③. MTA は宛先から送信先メールサーバーを探し, SMTP を用いて送信する④. 送信先 MTA は宛先が正しいか確認し受信する⑤. MDA がメールボックスにメールを格納する⑥. MUA が POP や IMAP を用いてメッセージを取得し処理を行う⑦⑧. 上野 (2005)[4] を参考に作成.

本論文の構成は以下の通りである. 2 章では電子メールの配送と安全な配送を実現する技術について整理する. 3 章では本研究の提案手法である MTA-STS を用いた電子メール通信経路の確立について, その目的と提案手法について述べる. 4 章では実装について言及する. 5 章では本研究の定性評価を行う. 最後に 6 章で MTA-STS の現状と結論を提示する.

2. 関連技術

本節では電子メール配送技術とその安全な配送を実現する技術について整理する.

2.1 SMTP

Simple Mail Transfer Protocol(SMTP)[5] は電子メール配送のインターネット標準プロトコルだ. SMTP は送信元からそのメールサーバー, 送信元メールサーバーから送信先メールサーバーとの通信に用いられる. 図 1 に SMTP を用いた電子メール配送の例を示した.

SMTP が考案されたのは 1981 年であり, その当時の SMTP には中継時のメッセージの秘匿性の保護や, 受信時のメッセージの認証の機能はなかった. そのためメッセージの傍受のような受動攻撃や, 改ざんやなりすましといった能動攻撃が可能であった. そうした攻撃に対応するため, 電子メールは拡張機能でメッセージの暗号化や送受信者認証を取り入れた.

次項では電子メールの配送・認証の 2 点に着目し, セキュリティ機構である STARTTLS, DKIM, SPF, DMARC について説明する.

2.2 配送での電子メールの保護

2.2.1 STARTTLS

STARTTLS[6] は “SMTP service extension for secure SMTP over transport layer security” として RFC3207 で定義されている SMTP 拡張機能だ. STARTTLS は個々の MTA 間の通信を TLS を利用して行うことで, SMTP のトラフィックを暗号化する. STARTTLS の機能は受動攻撃者の盗聴から通信を秘匿することだ. STARTTLS は通信に SMTP のウェルノウンポートを利用する点や, 暗号化に対応していない相手とも通信が可能な点から, 拡張機能として広く採用されている.

一方で, STRATTLS が責任を持つのは個々の中継であり, 全ての経路の通信が TLS 上で行われる保証はない. また STARTTLS では TLS 接続を確立する最初のハンドシェイクが平文で行われるため, その通信が改ざんされると TLS 上での通信を行うことができない. そしてドメインの認証を行わないため, DNS 偽装が行われた場合, 全ての配送で TLS 接続を保ったまま, 中間者攻撃を成立させられる危険性を有する.

2.3 認証を用いた電子メールの保護

メールサーバーは受信メールの整合性を検証するため, 補完的機能を有する. STARTTLS はサーバー間の個々の通信について暗号化を行ったが, 本項では電子メールの認証の観点からセキュリティ技術について言及する. またその動作フローについて図 2 に示した.

2.3.1 DKIM

RFC6376 で定義されている DomainKeys Identified Mail(DKIM)[7] は, 送信された電子メールの送信者ドメインが正しいか, 途中で改ざんされていないかメッセージの一貫性を認証する仕組みだ. 送信元サーバーはドメインに紐づいた秘密鍵でメールに電子署名を行い, メッセージヘッダーに `DKIM-Signature` を付与する. 送信先サーバーでは公開鍵を DNS サーバーの TXT レコードから取得し, メッセージの署名の検証を行う. DKIM そのものは, 認証の失敗時や, 署名がなかった場合の動作を規定しない.

2.3.2 SPF

Sender Policy Framework(SPF)[8] は RFC7208 で定義されている. SMTP はその仕組みから, 差出人メールアドレスを詐称することが容易だ. SPF は DNS を用いてこれを防ぐ. 送信元サーバーは対応するドメインを DNS の

SPF レコードとして登録し, 送信先サーバーはその DNS レコードを元に認証を行う. SPF は規定されたドメイン外からの電子メールの配送を拒否する.

2.3.3 DMARC

DMARC は“Domain-based message authentication, reporting, and conformance”として RFC7489 に定義されている [9]. DMARC は SPF, DKIM の両者が認証に失敗した場合の動作を規定するプロトコルだ. SPF, DKIM はその認証が失敗した際, その結果からどのように電子メールを取り扱うかの規定が曖昧であり, その判断は個々のサービスに委ねられていた. DMARC はその認証に失敗した際のポリシーを定義するとともに, 認証結果を提示する. DMARC レコードは `_dmarc.domain.com` のように規定された DNS サーバーの TXT レコードに記載される.

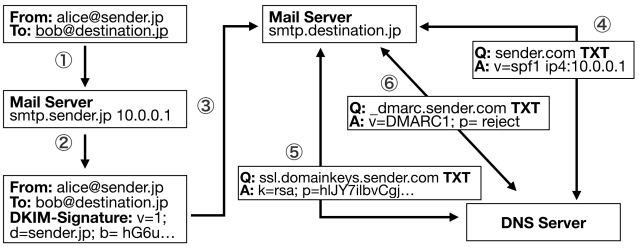

図 2 電子メール認証の概要 — SPF, DKIM, DMARC はそれぞれ送信者の認証を行うセキュリティ技術だ. 送信元メールサーバーはメールに電子署名を行う②. 送信先メールサーバーは SPF のホワイトリストの確認をする④. DKIM 署名の検証を行う.⑤. SPF, DKIM の両者の認証が失敗すると, DMARC レコードを問い合わせ処理を行う⑥. Zakir(2015)[2] を参考に作成.

3. MTA-STS を用いて暗号化された電子メール通信経路の確立

本節では本研究の背景, 目的とその提案手法について述べる.

3.1 STARTTLS の限界

前述したように SMTP には多数のセキュリティ技術が施されている. しかしながら, SMTP の通信経路の暗号化を担う STARTTLS には限界がある. STARTTLS を用いた場合の問題点は, 以下の 2 点に集約される.

- 通信の全ての暗号化は担保されない. 特にコネクションを確立するはじめのハンドシェイクは, 平文で行われる.
- プロトコルがドメインの証明を行わない.

この問題は STARTTLS が日和見暗号にあることに起因しており, 能動攻撃である中間者攻撃に対し脆弱である. 中間者攻撃の例として以下のものが挙げられる.

- **POODLE 攻撃**
  Padding Oracle On Downgraded Legacy Encryption(POODLE 攻撃) は TLS 1 系が普及した現在でも多くのサーバーが脆弱な SSL3.0 との互換性を保持していることを利用した中間者攻撃だ. 現在のトランスポート層の暗号化技術は TLS1.2 が採用されているが, 互換性を持たせるため, ダウングレードしての通信も可能である. この機能を悪用し, パディングオラクル攻撃に脆弱な SSL3.0 で通信を行わせ, 中間者攻撃を行う手法だ. [10]
- **DNS Cache Poisoning**
  STARTTLS は通信の暗号化を担保するが, その通信先ドメインについての認証を行わない. そのため DNS Cache Poisoning が行われた場合, TLS 通信を保ったまま, 意図しない接続先と通信をする危険性がある.

3.2 研究目的

前述の問題から, 本研究では, この STARTTLS の中間者攻撃への脆弱性に着目し, SMTP MTA Strict Transport Security (MTA-STS) を用いた電子メール通信経路の暗号化を提案する. また同時に MTA-STS の運用の実態の調査を行った.

3.3 MTA-STS

3.3.1 MTA-STS の概要

SMTP MTA Strict Transport Security (MTA-STS)[3] は IETF UTA Working Group で審議中の STARTTLS を用いたセキュリティ機構だ. MTA-STS の機能的特徴は以下の 3 点だ.

- どの MTA からの配送でも TLS が利用される.
- ドメインの MX ホストが正当性を提示する.
- TLS が利用できなかった際の振る舞いを定義する.

MTA-STS はその認証に DNS の TXT レコードと, JSON ドキュメントを用いる. 図 3 にその認証のフローを図示した.

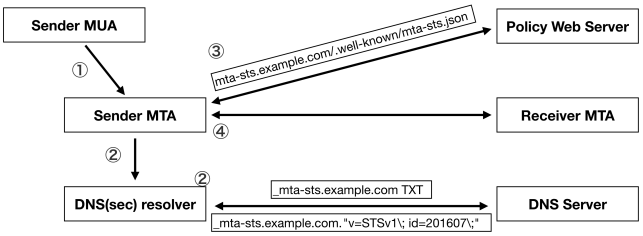

図 3 MTA-STS の概要 — MUA が送信先 MTA にメールを配送する①. 送信先 MTA は DNS に MTA-STS TXT レコードを要求する②. 送信先 MTA は Web サーバーに JSON で記述されたポリシーを要求する③. STARTTLS 接続をし, 認証に成功したらメールの配送を行う④. Men & Mice(2017)[11] を参考に作成.

また MTA-STS は状況に応じて, 2 種類の方式を用意し

ている.

- **report** モード
  このモードでは認証に失敗した際でも, メールが送信される. また認証に失敗した際は, 指定されたアドレスにエラーレポートを送信する.
- **enforce** モード
  このモードでは MTA-STS ポリシーでの配送を強制する. 認証に失敗した場合は, メールの配送を行わない.

3.3.2 MTA-STS TXT レコード

MTA-STS の TXT レコードはドメインが `example.com` であれば, `_mta-sts.example.com` のようなポリシードメインの形で格納される. TXT レコードはバージョンを表す “v”と, 暗号化のポリシーが記載された “id”によって構成される. 下に一般的な TXT レコードを示す.

```
_mta_sts IN TXT ("v=STSv1; id=20160707T010757\;")
```

3.3.3 MTA-STS JSON ポリシー

MTA-STS のポリシーは HTTPS 経由で提供される JSON ファイルに記述されている. JSON ファイルは以下のオブジェクトと値のペアを持つ. ポリシーの定義を以下に記す.

- **version**
  MTA-STS のバージョンが記載される. 現在は “STSv1” のみが提供されている.
- **mode**
  “enforce” または “report”が指定される.
- **mx**
  ドメインの MX パターンが記される.
- **maxage**
  ポリシーの有効期間が指定される.

JSON ポリシーの記述例を下に示す.

```
{
    "version": "STSv1",
    "mode": "report",
    "mx": ["*.mail.example.com"],
    "max_age": 86400
}
```

4. 実装

本提案手法の有用性を確認するため, MTA-STS を JavaScript を用いて実装した. 実装は MTA-STS のインターネットドラフト [3] に基づき, **report** モードでの配送が可能な設計を行った.

4.1 実装環境

実装の環境は表 1 の通りだ. 実装と実験は CentOS 6.9 上で行った. 実際のメール配送には MTA として Exim を用いた. MTA-STS の認証は, Node.js で動作するウェブサーバーが行う. クライアント証明書は Let’s Encrypt を用いて作成した.

表 1 実装環境

	使用した技術等	バージョン
OS	CentOS	6.9
MTA	Exim	4.89
実行環境	Node.js	64
証明書	Let’s Encrypt	—

4.2 システム構成

下にシステム構成図を示す (図 4). 送信者がメールを作成し送信すると, 認証のために Node.js サーバーが中継を行う①. Node.js サーバーは DNS サーバーから, MTA-STS レコードを取得する②. Node.js サーバーはポリシーが記載された JSON を送信先ウェブサーバーから HTTPS 経由で取得する③. Node.js サーバーは認証局に, 証明書の正当性を確認する④. メールが認証された後, Node.js は送信元 MTA にメールを中継し, メールが配送される⑤⑥. 正常に認証が行われた際は, メールヘッダーに `MTA-STS-Verify-True` と付加する.

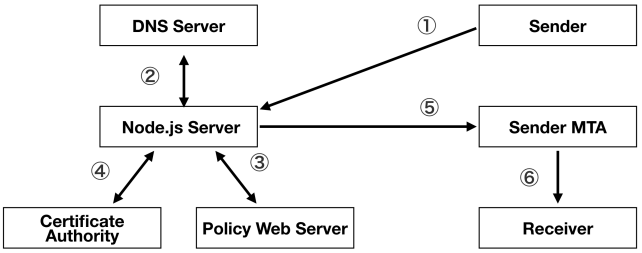

図 4 本システムの構成図

5. 評価

本章では, 本研究の MTA-STS 実装について, 既存メールサービスとの電子メール配送, インターネットサービスプロバイダとの互換性の 2 点について定性的に実施した評価について述べる.

5.1 既存メールサービスとの電子メール配送

本実装を用いて, 既存のメールサービスとのメール配送の実験を行なった. 今回の実験では MTA-STS ポリシーをサポートする `gmail.com` にメールを送信した. またポリシーを保持しないドメインにもメールを送信した. 結果, ポリシーを保持する `gmail.com` に送信した際にはヘッダーが付加され, そうでないドメインでは, ヘッダーが付加されないことを確認した.

5.2 インターネットサービスプロバイダとの互換性

本研究の実装が正しく機能していることを確認するため, 既存 ISP(インターネットサービスプロバイダ) のメールサーバーとの間で相互接続を実施した.

はじめに, 電子メールサービスを提供する主要な ISP[12] について, MTA-STS への対応状況について調査を実施した.

表 2 主要 ISP の MTA-STS 対応状況

ドメイン名	TXT レコード	ポリシー
comcast.net	○	report のみ
gmail.com	○	report のみ
gmx.com	○	report のみ
googlemail.com	○	report のみ
google.com	○	report のみ
mail.com	○	report のみ
yahoo.com	○	report のみ
gmx.net	○	report のみ
fastmail.fm	×	ポリシーなし
web.de	○	report のみ
mail.ru	×	ポリシーなし

調査では, インターネットドラフト [3] で MTA-STS にサーバーが対応しているかを確認するために定義された DNS TXT レコードによるサービス発見ならびに MTA-STS ポリシーの取得により対応状況を調査した.

表 2 は, 調査した主要 ISP ドメイン (102 個) のうち, TXT レコードが存在したドメインについて記す. fastmail.fm や mail.ru は TXT レコードは存在したが, ドラフト [3] で定める MTA-STS フォーマットに準拠していなかった. (いずれのドメインも SPF で使用する TXT レコードと同様のレコードが設定されていた.) TXT レコードが存在した 9 ドメインについては, MTA-STS ポリシーの有無について調査した. いずれのドメインも MTA-STS の動作モードは report モードであり, より厳格なポリシー適用を目指す enforce モードであるドメインは存在しなかった.

6. まとめ

本研究では, STARTTLS の持つ中間者攻撃への脆弱性に着目し, 対応策として策定中である MTA-STS を実装しその有用性を検証するとともに, 現在の運用状況について調査を行なった.

その結果, 本実装により MTA-STS を用いた電子メール配送が行えること, MTA-STS のポリシーの運用が確認された全てのメールサービスがその運用に `report` モードを使用していることが判明した. `enforce` モードが使用されていない理由としては, より厳格なこのモードでのメール配送が, MTA-STS の普及率から現実的でないからだと考えられる.

今後の課題として, report モードであると本来の目的の全ホップでの TLS 通信が達成されていないこと, また, enforce モードの普及とともに, どのようにメールサービスを運用するべきかについて考慮していくことが求められる.

参考文献

[1] Malatras, A., I. Coisel, and I. Sanchez. “Technical recommendations for improving security of email communications.” Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2016 39th International Convention on. IEEE, 2016.

[2] Durumeric, Zakir, et al. “Neither snow nor rain nor mitm...: An empirical analysis of email delivery security.” Proceedings of the 2015 ACM Conference on Internet Measurement Conference. ACM, 2015.

[3] D. Margolis et al, “SMTP MTA Strict Transport Security (MTA-STS).” Internet Draft, February 2017. `https://tools.ietf.org/html/draft-ietf-uta-mta-sts-03`

[4] 上野宣. 『今夜わかるメールプロトコル』. 翔泳社, 2005

[5] Postel, Jon. “Simple mail transfer protocol.” Information Sciences (1982). `https://www.rfc-editor.org/rfc/rfc5321.txt`

[6] Hoffman, Paul. “SMTP service extension for secure SMTP over transport layer security.” (2002). `https://tools.ietf.org/rfc/rfc3207.txt`

[7] Crocker, Dave, Tony Hansen, and Murray Kucherawy. “DomainKeys Identified Mail (DKIM) Signatures.” (2011). `https://tools.ietf.org/html/rfc6376`

[8] Kitterman, Scott. “Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1.” (2014). `https://tools.ietf.org/html/rfc7208`

[9] Kucherawy, Murray, and Elizabeth Zwicky. “Domain-based message authentication, reporting, and conformance (DMARC).” (2015). `https://tools.ietf.org/html/rfc7489`

[10] Möoler, Bodo, Thai Duong, and Krzysztof Kotowicz. “This POODLE bites: exploiting the SSL 3.0 fallback.” Security Advisory (2014).

[11] Men & Mice. “Webinar - SMTP STS (Strict Transport Security) vs SMTP with DANE.” `https://www.menandmice.com/resources/webinar-smtp-sts-strict-transport-security-vs-smtp-with-dane` (2017-5-10 参照).

[12] Zoho Corporation. “PublicEmailDomains” `https://www.zoho.com/salesinbox/help/zoho-crm/web/PublicEmailDomains.txt` (2017-5-10 参照).