送信ドメイン認証技術
導入マニュアル

第3.1版　2023年2月

迷惑メール対策推進協議会

目次

はじめに

現在のメールシステムは，その仕組みの簡便さもあり多くのコミュニケーションツールの基盤として広く普及しました．その一方で，メールを読むべきかを判断する，重要な情報である送信者を確認するための手段が備わっていなかったことにより，多くの問題が引き起こされています．例えば，送信者の確認ができないような匿名の広告宣伝メールや，実在する事業者を騙って偽のウェブサイトへ誘導することで，個人情報や金銭につながる情報を窃取するフィッシングが，大きな社会問題となっています．こうした迷惑メールは，メールに示される送信者情報が信用できないことから，受け取るべきかどうか，本当の事業者であるかどうかを判断ができないことが，大きな原因となっています．

送信ドメイン認証技術は，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るかどうかをドメイン名単位で確認（認証）する技術です．この技術により，送信者情報が詐称されることによって発生する多くの問題を解決できると期待されています．

送信ドメイン認証技術には，認証する送信者情報や認証に用いる仕組みが異なる複数の方式があります．送信ドメイン認証技術を利用するためには，メールの送信側と受信側のそれぞれで，新たな設定や機能の導入が必要となります．この導入マニュアルは，主にメールシステムの管理者やメール仕様の検討や導入を計画する立場にある企画担当者などを対象に，メールシステムの仕組みや各役割についての説明，送信ドメイン認証技術の導入にあたって必要な事項をまとめています．今回の導入マニュアルの改訂にあたっては，第 2 版以降に仕様が作られた DMARC を中心に，これまでの SPF と DKIM との関係も含めて解説しています．基礎編では，電子メールの基本的な仕組みと課題，SPF，DKIM，DMARC それぞれの送信ドメイン認証技術の技術仕様，関連するメールヘッダの仕様について解説します．応用編では，これら送信ドメイン認証技術を導入する際に検討しておくべき事柄について，DMARC を導入することを前提に解説します．送信ドメイン認証技術は，メールの送信側と受信側双方での新たな設定や機能の導入が必要となりますので，メールの送信側と受信側それぞれの立場で解説します．また，巻末に参考情報や用語の索引も付属しました．

送信ドメイン認証技術は，あくまでドメイン名単位で送信者情報が正しく設定されているかを確認（認証）する技術であり，この技術だけで送受信されるメールが迷惑メールかどうかを判断できるわけではありません．しかし，この技術の活用により送信ドメイン名の詐称ができなくなれば，ドメイン名単位で受け取りたいメール，受け取る必要がないメールを確実に判断できるようになります．送信ドメイン認証技術が広く普及することで，このような受け取り判断が可能なメールが増え，それにより，迷惑メール自体も減少することが期待できます．本導入マニュアルが，メールシステムの健全な発展に貢献できることを願っています．

迷惑メール対策推進協議会 技術ワーキンググループ
主査 櫻庭 秀次
副主査 加瀬 正樹
北崎 恵凡

第 1 章

基礎編

1.1　メールの仕組みと課題

本節では，送信したメールが受信者に届くまでの仕組みについて解説します．これにより，迷惑メールがなぜ受信者に届いてしまうのか，それを受信側で防ぐことが難しい理由について，メール配送の仕組みとともに解説します．

1.1.1　メールシステムの概要

メールの送信者が作成したメールは，インターネット上の幾つかのメールサーバを経由して受信者に届けられます．メールサーバは，メール送信者からの接続要求を受けて，メール配送上の決められた手順によってメッセージを受け取るサーバシステムです．メールの作成者は，投稿用のメールサーバにメールを送信し，メール受信者はメールの保存サーバからメールを取得しますが，その間には幾つかのメールサーバが介在します．図 1.1 に，example.com ドメイン名を利用している送信者から，example.jp ドメイン名を利用している受信者へメールが届けられるまでの流れを示します．

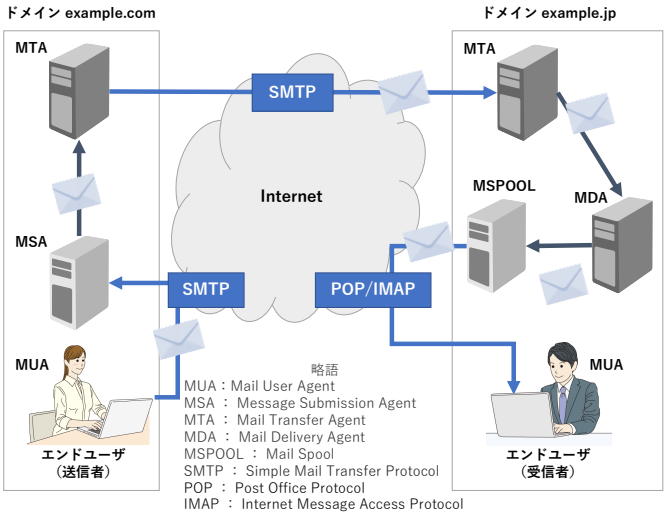

図 1.1　インターネットでのメール送受信

図 1.1 の説明

Mail User Agent （MUA: メールユーザエージェント）

メール利用者が，メッセージを作成し送信する（MSA にメールを投稿する）機能や，受信したメールをエンドユーザが読む（メールスプールから読み出し，表示する）ための機能を提供します．MUA から MSA へのメールの送信には，一般に SMTP プロトコルを利用します．また，メールスプールからの読み出しは，Post Office Protocol 3(POP 3) や Internet Message Access Protocol 4(IMAP 4) プロトコルを利用します．一般的にエンドユーザの端末上で動作するアプリケーションとして実装され，メールクライアント，メールソフト等と呼ばれます（例: Microsoft 社の Outlook や Mozilla の Thunderbird 等）．

Mail Submission Agent （MSA: メールサブミッション（投稿）エージェント）

エンドユーザにより作成されたメールを MUA などから受信し，メールの配送を開始する機能を提供します．MTA の役割をもつサーバアプリケーションプログラムが MSA を兼ねる場合もあります．MSA から MTA へ SMTP プロトコルを利用してメールを送信すると，MTA は次の送信先（他の MTA や MDA）へメールを配送します．

Mail Transfer Agent （MTA: メール配送エージェント）

メールを異なるメールシステム間で転送する役割を持ち，他の MTA や MSA と SMTP プロトコルなどを利用してメールの送受信を行う機能を提供します．メールの宛先に応じて転送する先を振り分ける機能や，受信側のシステムで受信する準備ができていない場合などに一時的にメールを保存（キュー）する機能などを持ちます．一般にサーバアプリケーションプログラムとして実装されます（例: sendmail, Postfix 等）．

Mail Delivery Agent （MDA: メール配送エージェント）

メールの受信側（宛先）で，MTA からメールを受信し，メールスプールへメールを保存する機能を提供します．MTA の役割をもつサーバアプリケーションプログラムが MDA を兼ねる場合もあります．

Mail Spool （MSPOOL: メールスプール）

メールの受信側（宛先）で，エンドユーザが MUA を利用してメールを読み取るためにメールを保存する機能を提供します．Mail Box と呼ぶ場合もあります．

以下にメールが配送されるまでの流れを示します．

1. ドメイン名 example.com の利用者が MUA を利用してメールを作成
2. 利用者は MUA を操作し，MSA にアクセスしてメールを投稿（送信）
3. MSA は，受け取ったメールを MTA へ配送
4. MTA は，宛先メールアドレスのドメイン名から DNS を参照することで受信 MTA の情報を取得し，メールを配送
5. 宛先ドメイン名の受信 MTA は，内部の MDA へメールを配送
6. MDA は，メールスプールへメールを届け，メールスプール領域にメールを格納
7. 宛先の利用者は，MUA を利用してメールスプールへアクセスし，自分宛のメールを読み出す

3，4，5 のメールの配送では，複数の MTA を経由する可能性があります．最近では，アンチウィルスフィルタや迷惑メールフィルタ，各種機能を提供するメールサービスも多く，メールサービス内部では各種機能を

提供する MTA を経由して配送される場合もあります．また，4 で宛先ドメイン名の MTA を探すときに，送信元の MTA は，宛先ドメイン名の DNS 上の MX レコードを参照します．MX レコードとして登録されているホストは，そのドメイン名において外部ドメイン名からのメールを受信する役割をもつホストと解釈されます．

DNS(Domain Name System) は，インターネット上でドメイン名を管理運用するためのシステムです．ドメイン名とは，管理の主体を階層的にドット（.）で結合した名前のことで，その名前に関連したインターネット上の様々な情報（IP アドレスや付随する情報など）は，DNS に問い合わせることで取得することができます．ドメイン名を横書きした場合，より上位の階層が右側となるように表現され，最上位のルートがドット（.）となります．ルートの次の名前（ラベル）が TLD(トップレベルドメイン）です．`example.jp.` の TLD は `jp` となります．また，このようにルートからの階層を省略せずに全て表記したドメイン名を，FQDN(Fully Qualified Domain Name) と示す場合があります．この関連情報の種類をあらかじめ決めたものが，資源レコードであり，IANA(Internet Assigned Numbers Authority) によって管理されています．DNS の資源レコードには，IP アドレスを示す A レコード (IPv4) や，AAAA レコード (IPv6)，メールサーバのホスト名を示す MX レコード，汎用的に情報を設定できる TXT レコードなどがあります．

1.1.2 メール配送の手順

メールを宛先に届けるための配送手順として，SMTP(Simple Mail Transfer Protocol, RFC5321) が使われます．メール配送では，メールを送りたい側（送信側）が，メールを届けたい先（受信側）にネットワーク的に接続を要求し，受信側が要求を受理することにより，一連の配送手続が開始されます．そして，実行したいコマンド名とその引数を相手方に送信し，相手方がそれに対する応答を返信するやりとりを繰り返すことにより手続が進みます．なお，メール送信は，送信側から受信側へ処理を伝えるので，ほとんどの場合，送信側が受信側に実行したいコマンドを送信することになります．

このときの接続の方法やメールの受け渡し，受信側からの応答の仕方などを決めたものが SMTP です．SMTP での手続の一般的な流れを図 1.2 に示します．

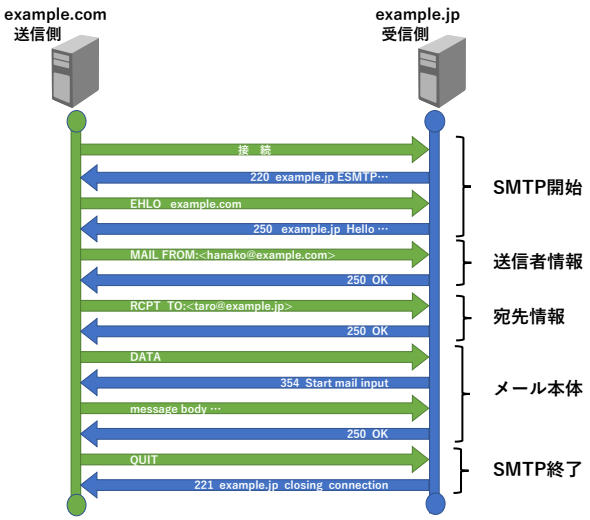

図 1.2　SMTP によるメール配送手順

送信側では，処理の内容を示すコマンド名（MAIL，RCPT，DATA など）を，引数とともに受信側に伝えます．例えば，MAIL FROM で送信者の情報（メールアドレス）を，RCPT TO で宛先の情報（メールアドレス）を，DATA でメール本体の情報を伝えます．受信側では，それぞれのコマンドを解釈して応答をします．コマンドに対する応答の種別は，数字（220 や 250 などの 3 桁の数字）で示されます．例えばメールを受け取れない場合には，否定的な応答番号（500 番台の数字）を返すことになります．受信側は，このようなコマンドに対する応答という形で，受信側の判断や意図を送信側に伝えることができます．SMTP の規格（RFC5321）では，送信側が指定する各コマンドの形式や順番，それぞれのコマンドに対して受信側が返すことができる応答番号の種類などが定められています．

メールの配送時に指定される，送信者や宛先のメールアドレスなどは，郵便の手紙になぞらえてエンベロープ（封筒）情報とも呼ばれます．そのため，MAIL コマンドで指定する送信者情報は，エンベロープ From と表現される場合があります．この SMTP は，メールサーバ間での配送だけでなく，利用者がメールを作成する MUA（メールソフト）から MSA（メール投稿サーバ）への送信（投稿）時にも用いられるプロトコルです．

1.1.3　メール本体の構造

メール本体は，送信側が DATA コマンドを送信し，受信側がその応答として 354 を返した場合に，送信側から送信されます．メール本体の最後に，ドット（.）だけの行（ドット行）が送信されます．なお，この場合のドット行は，メール本体には含まれません．メール本体は，ヘッダ領域とメール本文によって構成されます．いずれの領域もテキスト文字列で表現され，専用の区切り記号は決められていませんが，文字列の構成方法によって区別ができるように決められています．ヘッダ領域は，メール本体の先頭からメールヘッダ行が連続する領域で，メールヘッダ行ではない空行（改行だけの行）がメール本文との区切りとなります．

メール本文は，ヘッダ領域の区切り（空行）から，メール本体の末尾までとなります．メール本体の例を図 1.3 に示します

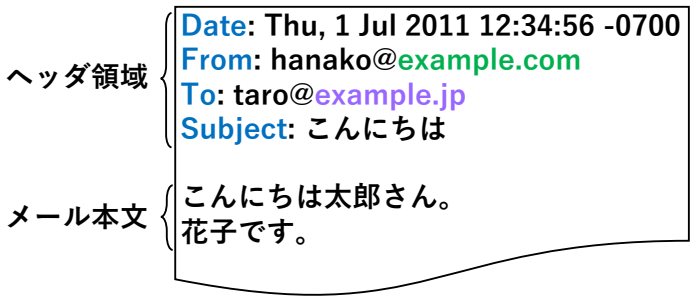

図 1.3　メール本体の例

メールヘッダ行は，ヘッダ名と続くコロン（:），ヘッダ本文で構成されます．メールヘッダ行はそれぞれ形式や構文が決まっており，メールヘッダを追加する場合には，正しく構文に従って記述しなければなりません．なお，メール本体の構造及びヘッダ行の構文は，SMTP の規格（RFC5321）とは別に，Internet Message Format（RFC5322, インターネットメッセージの形式）で決められています．

1.1.4　メールの送信者情報

メールの作成者や送信者を示す送信者情報には，メールの配送時に指定される送信者（エンベロープ情報に含まれる送信者情報）と，メール本体のヘッダ領域に記述される送信者情報の 2 種類があります．これらの送信者情報は，メールで利用される情報なので，いずれもメールアドレスで示されます．

メールの配送時に指定される送信者情報は，図 1.2 で示した MAIL コマンドの引数として指定されるメールアドレスです．メール本体で示される送信者情報と区別して，エンベロープ（封筒）上の送信者であることから，エンベロープ From と呼ばれることもあります．このメールアドレスは，配送上で何か問題が発生した場合に送信者に通知を送る際に利用されます．そのため，規格上はリバースパス（reverse-path）という名称が付いています．さらには，規格（RFC5321）での送信者情報ということで，RFC5321.From と表現されることもあります．いずれの情報も同じものを指していますが，ヘッダ上の送信者情報と区別するための表現となっています．リバースパスは，SMTP によるメールの配送時に渡される情報なので，一般にはメールの受信者には渡されない情報です．

受信側の MTA は，この MAIL コマンドに対する応答として，メールを受け取らないことを示すエラーコードを返すことができます．例えば指定されたメールアドレスが明らかに実在しないものであった場合や，過去に迷惑メールを送信してきた送信者であった場合などには，受け取らないことを応答コードで示すことができます．また受信側の MTA が，一旦メールを受け取った後で，MDA(Mail Delivery Agent) 上で MSPOOL(Mail Spool) に保存ができなかった場合や宛先が実際には存在しなかった場合など，何らかの問題があった場合には，このリバースパス宛にエラーメールを送信することになります．エラーメールは，NDR(Non-Delivery Reprot) や DSN(Delivery Status Notification)，NDN(Non-Delivery Notification) などとも表現されます．エラーメールが送信先に届かない場合，再度エラーメールが送信されメールがループすることを防ぐために，エラーメールのリバースパスには，null[*1]が用いられます．

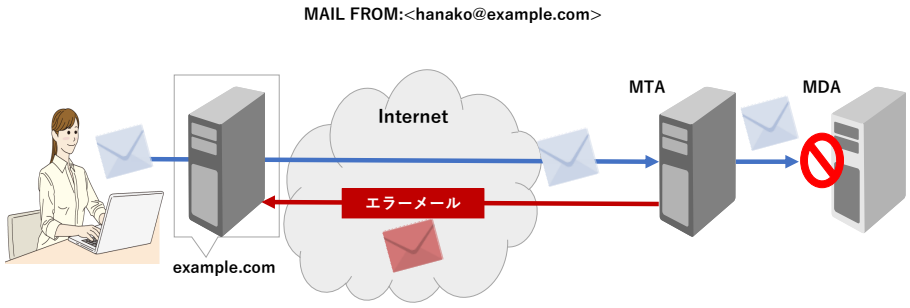

図 1.4　エラーメールの送信例

[*1] MAIL コマンドとしては MAIL FROM:<>と表現される．

ヘッダ領域は，メール本体の一部として受信者に届けられるため，受信者が見ることのできる情報です．しかしメールヘッダにはたくさんの種類があるために，受信者が利用する MUA（Mail User Agent, メールソフトウェア）の多くは，全てのメールヘッダを表示しません．送信者情報を示すヘッダの仕様としては表 1.1 に掲げるものがありますが，MUA が送信者情報として表示するのは，この From ヘッダが一般的です．このヘッダ上の送信者情報は，ヘッダ From やメールフォーマットの規格（RFC5322）から RFC5322.From とも表現されます．

表 1.1　ヘッダ上の送信元アドレス

ヘッダ名	用途
Sender	メール送信を実際に行った送信者を示す情報
From	メールの作成者を示す情報
Reply-To	返事を送信する場合の宛先を示す情報

Sender ヘッダが最も良く使われるケースとしては，メーリングリストがあります．メーリングリストでは，リストへの投稿者が From ヘッダに示されます．しかしリストメンバへの送信は，実際にはメーリングリスト機能を実現するプログラムなどが行います．そのため，Sender ヘッダにはメーリングリストの管理者などのメールアドレスが示されることになります．また，今ではあまり行われないかもしれませんが，例えば実際のメール内容の作成者に代わって秘書がメール送信するような場合に，送信者のメールアドレスが Sender ヘッダに示されることがあります．

From ヘッダの使われ方にも注意すべき点があります．From ヘッダには，メールアドレス以外にも，補助情報としてディスプレイネーム（display-name，表示名）を追加することができます．以下の例では，Email Sender 部分がディスプレイネームです．

ディスプレイネームの利用例

```
From: Email Sender <email-sender@example.jp>
```

ディスプレイネームには，任意の文字列を使用できますので，From ヘッダ上に示されたメールアドレスと全く関係の無い名前や文字列を記述できます．ディスプレイネームでは，よく二重引用符文字 (") で囲う使い方が多いようですが，仕様上は必須ではありません．また，メール受信側が同じような対応が可能であることが前提ですが，base64 等でエンコードすることで日本語など任意の文字列を記述することができます．しかしながら，ディスプレイネーム部分にメールアドレスを記述するような詐称手法もありますので，注意が必要です（2.3.3 を参照）．MUA や Web ブラウザでメールの送受信を行うウェブメールシステムの多くは，送信者情報として，このディスプレイネームを優先して表示しますので，送信者情報として表示される文字列については，実際の送信者とは全く関係が無い場合もありますので，注意が必要です．

1.1.5　送信者情報詐称とその対策

送信者情報詐称

メールシステムにおける送信者情報には，1.1.4 で示したように用途に応じた複数の種類があります．これまでは，いずれの送信者情報もそれが正しく設定されているかを判断する仕組みがありませんでした．もとも

との電子メールシステムは，インターネットが現在のような形で普及する以前から存在してきたシステムであり，メールの利用者が限られていた時代から少しずつ拡張されて使われてきたために，現在のように送信者情報が詐称されることを想定していませんでした．そのため，容易に送信者情報を詐称した電子メールが送信可能で，メールがインターネットの主要メッセージングツールとなりつつある頃から，こうした詐称メールが送られてきました．

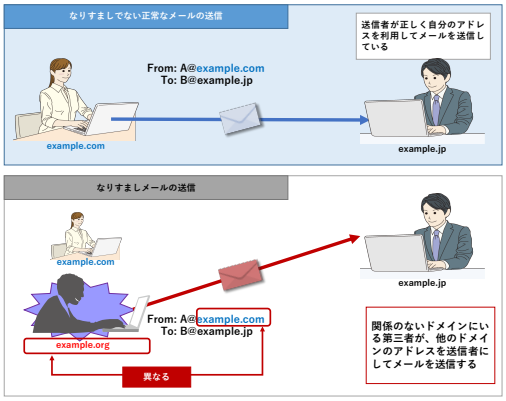

図 1.5　なりすましメール送信例

送信者情報詐称の問題点

メールの送信者情報が詐称されることによって，さまざまな問題が出てきています．まず，受信側で迷惑メールを受信しないために送信者情報をもとに受信を拒絶しようとしても，詐称されていることにより的確に受信の拒絶ができないことがあります．また実在する送信者を詐称し，実物とは異なる偽のウェブサイトに誘導し，個人情報を窃取するなどの犯罪行為（フィッシング）や，信頼性のある送信者を騙り，メールに添付された不正プログラムを実行させ，外部からコントロールされるボットにされたり，PC 内部にある個人情報や操作過程で得られる個人情報を窃取するなどの行為が行われています．

さらに，迷惑メールは宛先が存在するかどうかにかかわらず大量に送信される傾向にあるため，リバースパスを詐称し，実在するメールアドレスを指定している場合に，宛先が不明であることによるエラーメールが詐称されたメールアドレスに宛てて大量に送信される，バックスキャッタ (backscatter) が大きな問題となってきています．その場合には，そのエラーメールが迷惑メールと判断され，迷惑メール判定機能を提供しているベンダやブロックリストを運営している組織に報告され，正しい処理としてエラーメールを送信している側（最初のメールの受信側）が，迷惑メール送信者として登録され，同じ出口から送信される通常メールまでもが迷惑メール扱いされて届かなくなる，という 2 次的な問題も発生しています．

従来の対策手法と送信ドメイン認証技術

これまで受信側では，経験的な運用方法として，例えば送信者情報として存在しない（DNS で参照できない）ドメイン名が指定された場合や，日本の有名ドメインを利用しているが明らかに異なった地域の送信元から送信されたメールの場合に受信を拒否するなどの工夫をしてきました．しかしそれらの場合であっても，ドメイン名を管理する DNS(Domain Name System) が単に不調であったり，海外からメール転送をしていたりする場合など，ごくまれに送信者情報を故意に詐称したものではない正しいメールである場合があります．

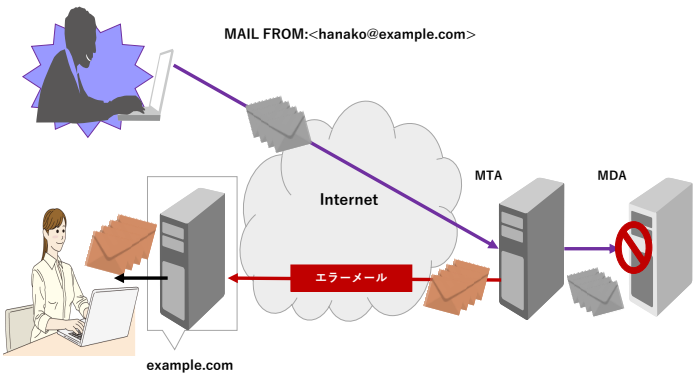

図 1.6　バックスキャッタ問題

このように送信者情報が簡単に詐称できてしまい，それを受信側で簡単に判断することが難しい現在のメールの仕組みが，大きな問題となっています．こうした問題に対応するために，送信ドメイン認証技術が開発され，これを普及させていくことにより，送信者情報の詐称の問題点を防ぐことが可能となります．これらの技術については，次節以降で詳しく説明します．

1.2 送信ドメイン認証技術によるなりすまし対策

送信ドメイン認証技術は，受信者が受け取ったメールの送信者情報が詐称されているかどうかをドメイン単位で確認（認証）する技術です．ここでは，送信ドメイン認証技術の概要について解説します．

送信ドメイン認証技術を利用すると，メールを受信したときに，それが本当に `From` ヘッダや `Return-Path` ヘッダに示されているメールアドレスのドメインから送られてきたかどうかを確認（送信ドメイン名を認証）することが可能になり，なりすましを見破ることができます．

送信ドメイン認証技術には，ネットワーク方式と電子署名方式という 2 つの異なる方式があり，さらにその 2 つを組み合わせた方式のものがあります．

ネットワーク方式の送信ドメイン認証技術では，SMTP プロトコルにおける送信元のホスト (MTA) の IP アドレスをもとにして認証します．この方式では，送信側で自ドメインで利用しているメールサーバ (MTA) の IP アドレス等を公開し，受信側で実際に通信している送信元の IP アドレスの妥当性確認を可能にします．この方式の認証技術には，Sender Policy Framework(SPF) があり，RFC7208 [*2] として仕様が公開されています．

電子署名方式の送信ドメイン認証技術では，公開鍵暗号技術を利用して認証します．この方式では，送信側で公開鍵と秘密鍵を用意し，あらかじめ公開鍵を公開したうえで，秘密鍵を用いて送信するメールに対する電子署名を作成し，メールに付与して送信します．受信側では，この電子署名を検証することで認証を行います．より具体的には，まずメール本文（特定のヘッダと本文）からハッシュ関数を用いてハッシュ値を計算します．ハッシュ値は，元のデータから得られる固定長のデータで，一般的には逆方向（ハッシュ値から元のデータを得る）の計算が困難であるという特徴があります．このハッシュ値を公開鍵暗号技術を用いて暗号化します．電子署名方式の送信ドメイン認証技術としては，DomainKeys Identified Mail (DKIM) がインターネット標準[*3]として公開されています．DKIM で利用できるハッシュ関数としては，SHA-1 あるいは SHA-256 が定義されています．

ネットワーク方式と電子署名方式を組み合わせた送信ドメイン認証技術としては，Domain-based Message Authentication, Reporting, and Conformance (DMARC) が情報提供のカテゴリ[*4] で公開されています．これらいずれの方式も既存のメールシステムの仕組みに大きな変更を加えなくても導入できるように考慮されています．

[*2] 当初は実験的 (Experimental) カテゴリの RFC4408 として 2006 年に公開されましたが，その後改訂され標準化への提唱 (Proposed Standard) カテゴリとなりました．

[*3] IETF が公開する RFC のカテゴリには，インターネット標準 (STD) に関するものと，それ以外に分類されます．インターネット標準 (STD) は，標準への提唱 (PS: Proposed Standard) を経て標準 (STD) となります．

[*4] インターネット標準 (STD) 以外のカテゴリとしては，情報 (Info: Informational), 実験 (Exp: Experimental)，歴史 (Hist: Histrical)，現状 (BCP: Best Current Practice) があります．

1.3 Sender Policy Framework (SPF)

Sender Policy Framework (SPF) は，ネットワーク方式の送信ドメイン認証技術です．IETF の MARID WG 等において数々の議論と検討を経て，実験的カテゴリの RFC(RFC4408) として規格化されたのち，現在は標準化への提唱として RFC(RFC7208) が公開されています．SPF では，リバースパス（RFC5321.From，エンベロープ From）を元に，認証を行います．

1.3.1 SPF の仕組み

SPF の認証の仕組みの概要を，図 1.7 に示します．

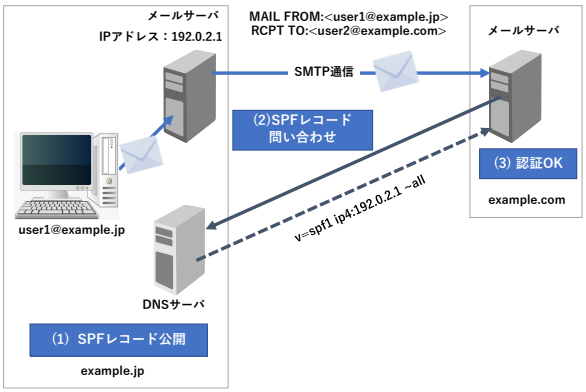

図 1.7　SPF による認証の仕組み

送信側では，あらかじめ自ドメイン名の DNS サーバ上に，自ドメイン名の送信者がメールを外部に向けて送出する可能性のあるメールサーバの IP アドレスの一覧を公開します（図 1.7 の (1)）．これを「SPF レコード」と呼びます．

受信側では，メールの受信時に送信者として指定されたリバースパス (RFC5321.From) のドメイン名部分に示されるドメイン名の DNS サーバより，SPF レコードを取得します（図 1.7 の (2)）．そして，その SPF レコードに指定されている IP アドレスと，SMTP で接続した先のメールサーバ（認証対象のメールサーバ）の IP アドレスが一致するか確認することで，認証を実施します（図 1.7 の (3)）．

1.3.2 送信側の設定

メールの送信側では，DNS 上で SPF レコードを公開するだけで SPF の運用を開始できます．RFC4408 では，SPF レコードは DNS の TXT 資源レコード (RR: Resource Record) か SPF RR として公開することが定められていましたが，RFC7208 では，DNS の TXT RR のみを利用することに変更されています．

SPF レコードには，当該ドメイン名に属するメールアドレスを送信者として，そのドメイン名外にメールを送信する可能性のあるメールサーバ (MTA) の，外向けの IP アドレスのリストを記述します．SPF では，IP アドレス（IPv6 を含む）を直接記述するほか，簡略に公開可能にする記述法が提供されています．SPF レコードの簡単な例を以下に示します．

SPF レコードの記述例

```
a.example.com.   IN  TXT  "v=spf1 -all"
b.example.com.   IN  TXT  "v=spf1 +ip4:192.0.2.1 -all"
c.example.com.   IN  TXT  "v=spf1 +ip4:198.51.100.1/28 ~all"
```

1 つめの例は，a.example.com をドメイン名として持つアドレス（例えば user@a.example.com）は，メールサーバ等の IP アドレスの記述が無いことから，全てのメールが SPF の認証が失敗 (fail) します．このことから a.example.com は，メールを送信しない (送信者情報として利用しない) ドメイン名であり，詐称に利用されないための SPF レコードを設定していると考えられます．

2 つめの例は，b.example.com をドメイン名として持つメールアドレス（例えば user@b.example.com）からのメールは 192.0.2.1 の IP アドレスを持つホストからのみ送信されるという意味を持ちます．

3 つめの例は，c.example.com をドメイン名として持つメールアドレス（例えば user@c.example.com）からのメールは 198.51.100.1/28 のネットワークのホスト（例えば 192.51.100.4 など）からのみ送信されるという意味を持ちます．

記述方法の詳細は，次項 (1.3.3) で解説します．

1.3.3 SPF レコードの記述法

SPF レコードは，最初にバージョンを記述し，空白（半角スペース）を入れて定義を記述します．定義は，空白（半角スペース）で区切り複数記述できます．

SPF レコードの定義

バージョン 空白 定義 空白 定義 ･･･（以下繰り返し）

バージョン

バージョン (Version) は，その SPF レコードが SPF のどのバージョンの文法にしたがって記述されているかを示します．RFC7208 で定義されている SPF レコードの文法はバージョン 1 のみであり，v=spf1 と記述することになっています．それ以外の記述（例えば v=spf1.0 等）をすると，その SPF レコードは不正なものとして，受信側で無視されることになります．

重要な点は，RFC で定められた文法に従わない誤った SPF レコードを記述すると，受信側ではその SPF レコードを無効として扱うということです (具体的には表 1.8 で解説する permerror として扱われることになります)．したがって，記述は慎重に行い，公開に際しては SPF レコードの記述について試験を行えるサイトなどを利用して，試験を行うことなどにより，記述ミスがないことの確認を推奨します．

定義

定義 (terms) は，ディレクティブ (directive) と修飾子 (modifier) で構成します．ディレクティブは，限定子 (qualifier) と機構 (mechanism) で構成します．

SPF の記述例の 2 つめでは，+ip4:192.0.2.1 と-all がディレクティブです．+ip4:192.0.2.1 のうち，+ が限定子で，ip4:192.0.2.1 が機構です．また，-all のうち，-が限定子で all が機構です．

修飾子の例としては，redirect=example.net 等があります．

受信側での認証の処理では定義は左から右へと評価します．認証対象のメールサーバの IP アドレスに対して最初に機構が適合した定義の限定子によって認証結果が決定されます．いずれの機構にも適合しない場合は，neutral と評価されます (neutral については，表 1.8 で解説します)．

SPF レコードの定義

定義 = ディレクティブ（限定子 + 機構）+ 修飾子

限定子

限定子 (qualifier) には，認証対象のメールサーバの IP アドレスが，それに続く機構に適合した場合の認証結果を指定します（SPF レコードの定義を参照)．

限定子には，+，-，～，?があり，+ 限定子は省略可能です．それぞれについての認証結果とその意味を，表 1.2 に示します．

表 1.2　SPF レコードの限定子

限定子	認証結果	意味
+	pass	当該ドメインの送信メールサーバとして認証する
-	fail	当該ドメインの送信メールサーバとして認証しない
～	softfail	当該ドメインの送信メールサーバとして認証しないが 正当なメールであっても認証失敗する可能性もある
?	neutral	認証されたかどうかを判断されたくない

機構

機構 (mechanism) には，認証対象のメールサーバの IP アドレスと照合する条件を記述します．機構には，all，include，a，mx などがあります．機構の種類とそれぞれの引数及び機能を，表 1.3 に示します．機構の多くは “:” や “/” の文字が名前のあとに利用できます．

SPF レコードに記述された全ての機構に適合しなかった場合は，SPF の認証結果は neutral となります．つまり，SPF レコードの末尾に?all と記述されていることと同等の扱いになります．

修飾子

修飾子 (modifier) は，認証についての付加的な情報を与えるものです．機構とは異なり，直接，認証結果を与えませんが，redirect のように認証結果に影響を与えるものもあります．修飾子のあとには = に続いてドメイン名が指定されます．引数のドメイン名は FQDN(Fully Qualified Domain Name) である必要があります．修飾子を表 1.4 に示します．

表 1.3　SPF レコードの機構

機構	引数	機能
all	なし	・すべての IP アドレスに適合する ・SPF レコードの末尾に置かれデフォルトの動作を定義するために利用される
include	ドメイン名	・引数に与えられたドメインの SPF レコードを使って認証処理を実施する ・その結果が pass，temperror 又は permerror の場合にのみ値が採用され，include に指定された先のドメインの SPF レコードによって fail の判定が与えられても，それが認証結果としては採用されない
a	ドメイン名	・認証対象のメールサーバの IP アドレスが，ドメイン名に対応する A (AAAA) レコードの IP アドレスのいずれかであれば適合する
mx	ドメイン名	・認証対象のメールサーバの IP アドレスが，ドメイン名に対応する MX レコードに指定されているホストの A (AAAA) レコードの IP アドレスのいずれかであれば，適合する ・MX は複数与えられる場合があるが 10 個までの MX ホストに対して検査する
ptr	ドメイン名	・認証対象のメールサーバの IP アドレスをリバースルックアップし，得られたホスト名でさらに正引きを実施して IP アドレスを得て、その IP アドレス（複数の IP アドレスが得られる場合がある）が送信元ホストの IP アドレスを含む場合でかつ，リバースルックアップによって得られたホスト名が ptr の引数に与えられたドメイン名と一致するかそのサブドメインである場合に，適合する ・リバースルックアップに失敗した場合は fail とみなすが負荷の多い処理となるため，あまり利用は推奨されない
ip4	IPv4 ネットワークアドレス (CIDR 表記可能) 又は IPv4 アドレス	・そのドメインのメールを送信する可能性のあるメールサーバの IPv4 ネットワークアドレス又は IPv4 アドレスを引数に指定する ・認証対象のメールサーバの IP アドレスが，指定された IPv4 ネットワークに含まれているか，IPv4 アドレスに合致する場合に適合する
ip6	IPv6 ネットワークアドレス (CIDR 表記可能) 又は IPv6 アドレス	・そのドメインのメールを送信する可能性のあるメールサーバの IPv6 ネットワークアドレス又は IPv6 アドレスを引数に指定する ・認証対象のメールサーバの IP アドレスが，指定された IPv6 ネットワークに含まれているか，IPv6 アドレスに合致する場合に適合する
exists	ドメイン名	・引数であるドメイン名に指定された表記で A レコード参照を実施し，該当の A レコードが存在すれば適合する ・SPF のマクロ機能とあわせての利用を想定する

マクロ

SPF レコードの評価中に，メッセージや接続時のパラメータに置き換えられる文字列を含むことができます．具体的には，送信者情報 (%{s}) やそのドメイン名 (%{d})，local-part 部分 (%{l}) や IP アドレス (%{i})，現在時刻 (%{t}) などの情報が利用できます．マクロは一見便利な機能ですが，これを多用すれば認証の過程を人が確認することが難しくなったりしますので，必要最小限の利用が良いでしょう．

SPF レコード公開時の制限

RFC7208 では，DNS 上の制約から DNS への SPF レコード問い合せ結果が 512 オクテットに収まるよう，SPF レコードの文字列は充分短くすべきとしています．この制約は，UDP を利用した DNS への対応に起因している他の多くの要因に依存しますが，ガイドラインとしては，ドメイン名とすべての対象テキストの合計が 450 オクテットより少なくすべきであることが示されています．また，DNS サーバの負荷や受信側の

表 1.4　SPF レコードの修飾子

修飾子	引数	機能
redirect	ドメイン名	・引数であるドメイン名に指定されたドメインの SPF レコードにより認証処理を実行する ・複数のドメインが 1 つの SPF 定義を共有するような場合での使用を想定する ・この修飾子を利用する場合には，SPF レコードの末尾に配置することを推奨する
exp	ドメイン名	・認証が失敗した場合に，引数であるドメイン名に指定されたドメインの TXT RR に設定されている文字列を，認証失敗した理由や説明等として利用する ・SMTP セッションでのエラーメッセージなどでの利用を想定する

認証処理を軽減する意味もあり，一度の認証処理で参照する DNS の回数の上限を 10 回に制限しています．これらについては，応用編でさらに詳しく解説します．

認証結果の扱い

認証結果については，1.6 で解説します．

1.4 DomainKeys Identified Mail(DKIM)

DomainKeys Identified Mail(DKIM) は，電子署名方式の送信ドメイン認証技術です．IETF において，STD76(RFC6376) として標準となりました．図 1.8 に示すように，DKIM では送信側のメールサーバで電子署名を作成して付与し，受信側のメールサーバでその電子署名を検証するという方法で送信者のドメイン名（署名ドメイン名）の認証を行います．電子署名はメール本体（ヘッダ及びメール本文の内容）をもとに作成するので，中継メールサーバ (MTA) などで何らかの理由で電子署名又は電子署名の元になったメール本体のデータが変更されなければ，たとえメールが転送されても，転送先で認証することができます．また，電子署名の情報はメールヘッダに記載しメール本文はそのまま維持されるため，MUA 等への影響もありません．

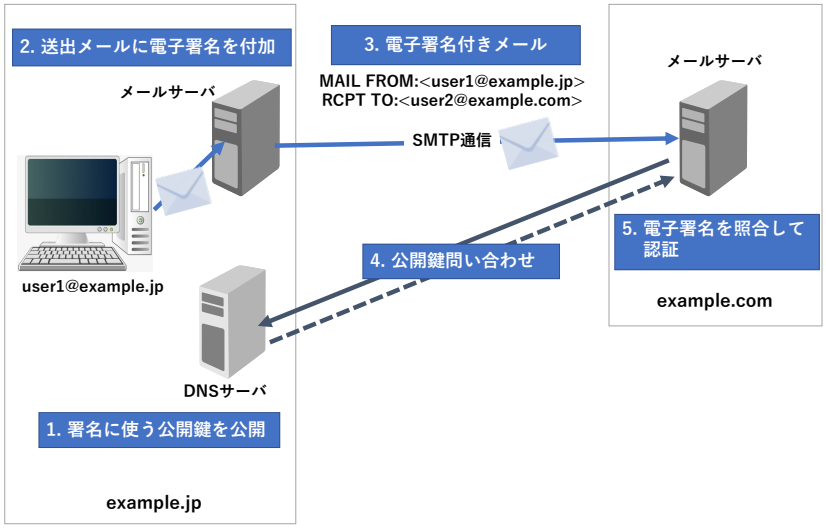

図 1.8　DKIM による送信ドメイン認証

1.4.1 DKIM の仕組み

DKIM の認証の仕組みの概要を，図 1.8 に示します．電子署名は公開鍵暗号技術を用いて作成されますので，秘密鍵と公開鍵の鍵ペアをあらかじめ用意しておく必要があります．作成された電子署名は，送信するメールのヘッダ (`DKIM-Signature`) のパラメータとして，タグ形式（タグ名＝タグ値）で示されます．`DKIM-Signature` で利用可能なタグについては，表 1.6 に示します．

DKIM では，次のような手順で DKIM 送信ドメイン認証を実施しています．

1. メール送信側の `example.jp` ではあらかじめ DNS に電子署名に使う公開鍵を公開しておく
2. `example.jp` のメールサーバでは送出メールの本文とヘッダを元に電子署名を付与する
3. メールを `example.com` のサーバに SMTP で送信する
4. メール受信側の `example.com` のメールサーバは `DKIM-Signature` ヘッダの `d` タグに指定されたドメイン部である `example.jp` の DNS へ公開鍵を問い合わせる
5. `example.jp` から取得した公開鍵により電子署名を検証し一致していれば認証成功

1.4.2 公開鍵の公開

DKIM では，送信側のドメインの DNS 上に，電子署名の作成に利用した秘密鍵に対応する公開鍵を公開します．公開鍵は，署名ドメイン名に対する TXT 資源レコード（DKIM レコード）として DNS に登録します．

多くの受信側のメールサーバでは，鍵の長さが 1,024 bit から 2,048 bit までをサポートしています．規格 (RFC) では 2,048 bit より長い鍵を利用する場合もあるとされています．なお，1,024 bit より短い鍵は，オフラインでの解読行為に対して脆弱ですので，利用を避けてください．

公開鍵を登録するドメイン名は次の通りです．_domainkey ラベルは，固定の文字列です．

公開鍵を登録するドメイン

```
<selector>._domainkey.<ドメイン名>
```

セレクタ (<selector>) は，DKIM-Signature ヘッダの s タグに指定したラベル（サブドメイン名）です．あるドメイン名に対して，サブドメイン名（ラベル）を複数設定することで，セレクタを複数用いることができます．これにより，同じドメインに対して複数の鍵ペアを運用することが可能になります．こうした運用は，メールの用途の違い等で複数の鍵ペアを同時に利用することが可能となり，例えばそれぞれで鍵長や暗号方式を変えるといった運用もできます．また，電子署名で利用する鍵ペア（公開鍵と秘密鍵）は，セキュリティの観点から定期的に鍵ペアの交換（ロールオーバー）をする必要があります．セレクタを利用することで，鍵ペアのロールオーバーをスムーズに実施できます．**<ドメイン名>**は，DKIM-Signature ヘッダの d タグに指定したドメイン名になります．

1.4.3 DKIM レコードの記述法

DKIM レコードの記述例を，以下に示します．ドメイン名は example.com で，セレクタ名は sls です．DKIM レコードが設定される TXT RR（テキスト資源レコード）は，汎用の資源レコードですので，必ず v=DKIM1 から始まるテキスト（文字列）である必要があります．

DKIM レコードの記述例

```
sls._domainkey.example.com.  IN TXT "v=DKIM1; k=rsa; t=y; p=MIGfMA0GCSqGSI...<省略>"
```

公開鍵のレコードは，各パラメータを「タグ名＝タグ値」のタグ形式をセミコロン (;) で区切って列挙して設定します．タグ名及びタグ値の前後，つまりタグ形式の前後には空白文字があっても構いません．利用できるタグ名を，表 1.5 に示します．

DKIM レコードで必須のタグは，公開鍵のデータ（p）だけですが，他の TXT RR を利用した DNS レコードと区別するためにも，バージョン番号（v）は設定したほうが良いでしょう．それ以外のタグ名は省略することができます．

表 1.5　DKIM レコードのタグ

タグ名	タグ値	説明	省略の可否
v	バージョン番号	・DKIM レコード形式のバージョン番号 ・指定する場合は DKIM1 とし，省略した場合も DKIM1 となる ・指定する場合はレコードの最初に記述する	設定することが推奨されるが省略可能
h	利用可能なハッシュ方式	・電子署名の作成の際に利用できるハッシュ方式 ・省略した場合にはすべてのハッシュ方式を許容する ・電子署名の作成者と検証者の両方が sha256 方式をサポートする必要があり，検証者は sha1 方式もサポートする必要がある	省略可能
k	鍵の形式	・電子署名の作成の際に利用できる鍵の形式 ・省略した場合には rsa となる	省略可能
n	説明	・可読な説明文を保持するタグ ・省略した場合には無し（長さ 0 の文字列）となる	省略可能
p	公開鍵データ	・公開鍵のデータを保持するタグ ・鍵データは base64 でエンコードする ・値が指定されない場合には該当の鍵が無効になっていることを示す	必須
s	サービスタイプ	・当該鍵が有効であるサービス ・コロン (:) で区切って複数指定できる ・現時点で指定できるサービスは，*（すべてのサービス）と email（電子メール）の 2 つがあり，省略した場合には*となる	省略可能
t	フラグ	・処理のオプションを示すフラグ ・コロン (:) で区切って複数指定できる ・指定できるフラグには y と s があり，y は DKIM の運用が試験モードであることを示す ・y が指定されている場合には受信者は認証に成功したメールと失敗したメールを区別して処理してはいけない ・s が指定されている場合は，i タグに指定されたアドレスの @ から右のドメイン名は d タグに指定された値と一致する必要があり，省略した場合にはフラグなしとなる	省略可能

1.4.4　送信側での電子署名の作成

送信側では，メール本体（ヘッダ及びメール本文の内容）をもとに電子署名を作成します．作成した電子署名の情報は，DKIM-Signature ヘッダとして付与します．DKIM-Signature ヘッダについても，通常のメールヘッダとして記述する必要があるため，メールヘッダの書式に準じた形式（一行の長さや複数行に折り返す場合の記述方法など）である必要があります．メールヘッダの書式でいえば，ヘッダ領域名（field name）は DKIM-Signature であり，コロン（:）に続いてヘッダ本体（field body）が続きます．DKIM-Signature ヘッダのヘッダ本体の書式は，「タグ名=タグ値」の組をセミコロン (;) で区切って列挙したタグ形式です．タグ名およびタグ値の前後には，空白文字があっても構いません．DKIM-Signature ヘッダの例を，図 1.9 に示します．

電子署名のデータやハッシュの値などは，本来はバイナリデータですが，メールヘッダの書式では，ヘッダ本体にはいわゆるアルファベットなど（US-ASCII）以外は利用できないため，バイナリデータは base64 でエンコードした文字列を設定します．電子署名にはメールヘッダも含めるため，含めるメールヘッダのヘッダ

領域名を h タグ名で指定します．この場合，指定する順番によって電子署名の値が変わったり，複数あるメールヘッダを含めないようにするなどの注意が必要です．

```
DKIM-Signature: v=1; a=rsa-sha256; s=brisbane; d=example.com;
    c=simple/simple; q=dns/txt; i=joe@football.example.com;
    h=Received : From : To : Subject : Date : Message-ID;
    bh=2jUSOH9NhtVGCQWNr9BrIAPreKQjO6Sn7XIkfJVOzv8=;
    b=AuUoFEfDxTDkHlLXSZEpZj79LICEps6eda7W3deTVFOk4yAUoqOB
    4nujc7YopdG5dWLSdNg6xNAZpOPr+kHxt1IrE+NahM6L/LbvaHut
    KVdkLLkpVaVVQPzeRDI009SO2Il5Lu7rDNH6mZckBdrIx0orEtZV
    4bmp/YzhwvcubU4=;
Received: from client1.football.example.com [192.0.2.1]
    by submitserver.example.com with SUBMISSION;
    Fri, 11 Jul 2003 21:01:54 -0700 (PDT)
From: Joe SixPack <joe@football.example.com>
To: Suzie Q <suzie@shopping.example.net>
Subject: Is dinner ready?
Date: Fri, 11 Jul 2003 21:00:37 -0700 (PDT)
Message-ID: <20030712040037.46341.5F8J@football.example.com>

Hi.

We lost the game. Are you hungry yet?

Joe.
```

出典： https://datatracker.ietf.org/doc/html/rfc6376

図 1.9　DKIM-Signature ヘッダ例

DKIM-Signature ヘッダで利用できるタグ名を，表 1.6 に示します．
送信側メールサーバは，次の手順で電子署名を作成し，DKIM-Signature ヘッダをメールに追加します．

1. 電子署名を作成する対象となるメールか確認する
2. 電子署名を作成する対象となるヘッダを決定し，h タグに列挙する
3. メール本文の内容から l タグに指定した長さを取り出し，正規化処理を実施する
4. 対象となるヘッダ，正規化したメール本文の内容，これから追加する DKIM-Signature ヘッダの電子署名のデータを取り除いた部分をつなげたデータに対して，ハッシュを作成する
5. ハッシュに対する電子署名を作成し，DKIM-Signature ヘッダの b タグの値として挿入したのち，DKIM-Signature 自体を対象メールのヘッダ領域に追加する

電子署名の対象とすべきメールヘッダには，From ヘッダがあります．また，次に示すメールヘッダもメール内容に関わるものとして例示されています．

```
Reply-To, Subject, Date, To, Cc, Resent-Date, Resent-From, Resent-To, Resent-Cc,
In-Reply-To, References, List-Id, List-Help, List-Unsubscribe, List-Subscribe,
List-Post, List-Owner, List-Archive
```

一方で，複数存在するヘッダやメール配送の途中で変更される可能性がある以下のヘッダは，電子署名の対象から除外すべきとされています．

```
Return-Path, Received, Comments, Keywords
```

表 1.6　DKIM-Signature ヘッダのタグ

タグ名	タグ値	説明	省略の可否
v	バージョン番号	・DKIM-Signature ヘッダ形式のバージョン番号 ・現時点では 1 と指定する	必須
a	電子署名の作成に利用したアルゴリズム	・電子署名の作成に利用したアルゴリズム ・rsa-sha1, rsa-sha256 と ed25519-sha256 が利用できる	必須
b	電子署名データ	・電子署名データ ・base64 にエンコードして指定する	必須
bh	本体のハッシュ値	・電子署名の対象とした本体のハッシュ値	必須
c	メール本文とヘッダの正規化方式	・電子署名の作成に利用する正規化処理の方法 ・スラッシュ (/) で区切って，それぞれメール本文の正規化に利用したアルゴリズムを表示する ・simple と relaxed が指定可能で，省略した場合には simple/simple となる	省略可
d	ドメイン名	・電子署名の作成を行ったドメイン名，すなわち送信ドメイン名 ・公開鍵の取得の際に参照するドメイン名の一部になる ・後述の i タグに与えられるアドレスのドメイン名は，このタグに与えられる値と同じか，又はサブドメイン名であることが必要である	必須
h	電子署名の対象としたヘッダ	・電子署名を作成するデータに含まれたヘッダ ・コロン (:) で区切って複数列挙できる ・送信者を示すヘッダである From ヘッダや Sender ヘッダなどは，必ず電子署名の対象に含めることが必要である	必須
i	認証対象送信者 アドレス	・メールの送信者や送信プログラム（メーリングリストなどの場合）のメールアドレス ・省略した場合には，d タグに指定したドメイン名の先頭にアットマーク (@) を追加した値になる	省略可
l	電子署名の対象とした本文の長さ	・電子署名の作成を行ったメール本文の先頭からの文字長（オクテット長） ・省略した場合はメール本文すべてを電子署名の対象とする	省略可
q	公開鍵取得方法	・公開鍵を取得する方法 ・現時点では dns/txt のみ指定可能，省略した場合には dns/txt となる	省略可
s	セレクタ	・鍵を選択するセレクタ ・公開鍵を取得する際に，クエリを発行する対象のドメイン名の一部に利用する ・複数のセレクタを持つことで，1 つのドメインで複数の公開鍵を利用できる	必須
t	電子署名実施時のタイムスタンプ	・電子署名を作成した日付 ・EPOC（UTC の 1970 年始め）からの秒数で指定，省略した場合は未定義となる	利用推奨（省略可）
x	有効期限	・電子署名の有効期限について，有効である日時 ・EPOC（UTC の 1970 年始め）からの秒数で指定，省略した場合は無期限になる	利用推奨（省略可）
z	電子署名の対象としたヘッダのコピー	・電子署名の対象にしたヘッダの電子署名の作成時の値 ・縦棒 (|) で区切って複数指定できる ・デバッグ目的であり，認証処理には利用しない	省略可

1.4.5 受信側での処理

受信側では，メールに付与された DKIM-Signature ヘッダを取り出し，次の手順で電子署名の検証を行います．送信側の公開鍵が取り出せなくなる場合があるため，電子署名の検証はできる限り受信時からあまり時間をあけずに実施すべきです．

1. DKIM-Signature ヘッダの d タグ，s タグの値から，公開鍵を取得するドメイン名を作成する
2. 作成したドメイン名をもとにして DNS から公開鍵等をもつ DKIM レコードを取得する
3. DKIM-Signature ヘッダの h タグに列挙されたヘッダと，l タグで指定された長さのメール本文の内容及び電子署名データ以外の DKIM-Signature ヘッダの値を合わせて受信したメールのハッシュを作成する
4. 公開鍵を利用して DKIM-Signature ヘッダの b タグが持つ電子署名からハッシュを復号する
5. 復号したハッシュと受信したメールから作成したハッシュを比較して，同一であれば認証成功とする

1.4.6 認証結果の取り扱い

DKIM では，認証対象のドメイン名はメールの From ヘッダではなく，DKIM-Signature ヘッダの d タグに指定されたドメイン名です．そのため，SDID(Signing Domain Identifier，署名ドメイン識別子) とも呼ばれます．このため，From ヘッダのドメイン名と認証に利用したドメイン名が異なる場合が存在します．一方で，DKIM には認証結果によって受信したメールをどのように取り扱うかについては定義されていません．

1.4.7 鍵ペアのロールオーバーと DMARC の併用について

DKIM の鍵ペアについて明示的に有効期限は設けられていません．しかし，DKIM を運用する場合は，セキュリティの観点で期間を定めて，新しい鍵ペアの運用に切り替えて，古い鍵ペアの破棄を行います．古い鍵ペアを破棄する前に，新しい鍵ペアでの運用が適切であるかを確認するため，1.5.5 で解説する DMARC の集約レポートを参考にします．詳細については，第 2 章の応用編で解説します．

1.5 Domain-based Message Authentication, Reporting, and Conformance (DMARC)

Domain-based Message Authentication, Reporting, and Conformance (DMARC) は, SPF および DKIM で認証されたドメイン名を利用して，メールヘッダ上の送信ドメイン名（RFC5322.From のドメイン名）を認証します．つまり，認証の基本的な技術は SPF や DKIM を利用しており，新たな認証のための仕組みを作り出したわけではありません．DMARC は SPF や DKIM の認証結果を利用して総合的に送信ドメインを認証する技術であり，既に SPF や DKIM の設定をしていれば，送信側での DMARC の導入は簡単にできます．DMARC では，SPF や DKIM の認証結果を利用し，双方の長所を活かした認証を行うことができます．また DMARC では，認証に失敗したメールの取り扱いを送信側でポリシー（DMARC ポリシー）として宣言できます．これにより，なりすまされているメールは受け取らない，といった強いポリシーを受信側に伝えることができるようになります．更に DMARC では，ドメイン管理者（メール送信側）が，ポリシー設定の判断を助けるために，メール受信側が送信する認証結果の報告（DMARC 集約レポート）の宛先を，DMARC ではレコードに設定することができます．

DMARC の仕様は，IETF で情報提供 (Informational) カテゴリの RFC7489 として規格化されています．現在は，標準への提唱 (Proposed Standard) を目指した改訂作業中であり，仕様が変更される可能性がありますが，ここでは，RFC7489 に基づいて DMARC の仕組みについて解説します．DMARC の仕様は，認証に関わる部分（DMARC レコードなど）とフィードバック（DMARC レポート）に関わる部分の二つに大きく分けられます．

1.5.1 DMARC の仕組み

DMARC は，SPF や DKIM の導入が前提となりその認証結果を利用します．原則として SPF または DKIM で認証されたドメイン名が From ヘッダ上の送信者（ヘッダ From）のドメイン名と一致しているかを確認し，両者の組織ドメイン名が一致している場合に DMARC として認証成功とします．メール送信側では次の手順で導入を行います．それぞれの番号は，図 1.10 に示した番号に対応します．

1. SPF を導入（送信ドメイン名の DNS で SPF レコードを設定，詳細は 1.3 を参照），DKIM を導入（送信側のメールサーバで認証に用いる公開鍵暗号技術による鍵ペア，公開鍵および秘密鍵を生成し，公開鍵を DNS サーバで DKIM レコードとして公開，詳細は 1.4 を参照）
2. 送信側の DNS で DMARC レコードを設定（詳細は 1.5.3，1.5.4 を参照）
3. DMARC レポートの受信準備をする（DMARC レコードにレポートの宛先を設定，詳細は 1.5.5 を参照）

メール受信側では，以下の手順で認証します．

4. DMARC 認証ドメイン名（識別子）の抽出
5. DMARC レコードを取得
6. DKIM 署名検証および署名ドメイン名の取得，SPF 認証の実行および認証ドメイン名の取得
7. SPF および DKIM の認証ドメイン名と DMARC レコードから識別子アラインメントを確認し，

DMARC の認証処理を行う

8. DMARC の認証結果と DMARC レコードのポリシー（DMARC ポリシー）から受信処理判断する
9. 受信時の認証結果をもとに DMARC レポートを生成し送信する

DMARC レポートを受信する設定にしている場合，メール送信側では次の対応を検討します．

10. 受信したレポートを参照し，正規のメールが認証失敗していないかを確認する
11. 正規のメールが認証失敗している場合には，失敗している SPF あるいは DKIM の設定を見直す
12. 正規のメールの認証失敗が発生していない場合は，送信しているメールの用途等を検討の上，DMARC ポリシーの強化を行う

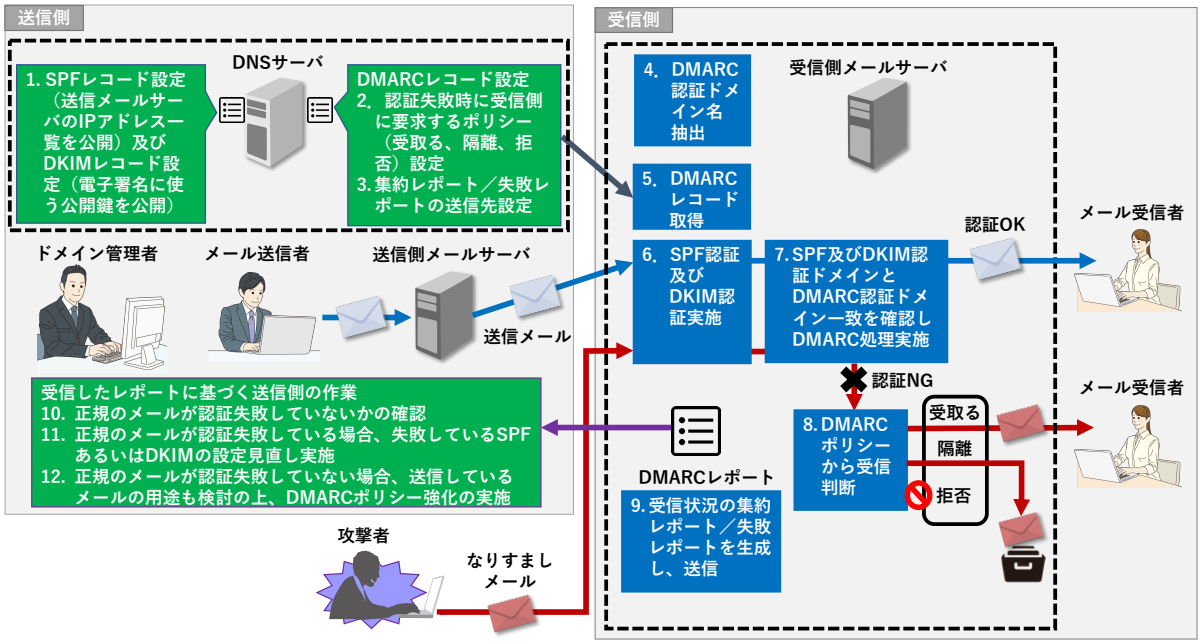

図 1.10　DMARC の仕組み

DMARC 認証ドメイン名は，メールヘッダ上（`From:` ヘッダ）の送信者ドメイン名です．メールフォーマットの形式を定めた規格，RFC5322 を用いて RFC5322.From と表現することもあります．通常 RFC5322.From は，一つのメールアドレスだけを設定しますが，規格上は複数のメールアドレスを設定することができます．この場合，どのドメイン名を認証すべきか判断できないため，DMARC としては認証対象外としています．

DMARC レコードは，対象ドメイン名のサブドメイン名 `_dmarc` の DNS TXT（テキスト）資源レコードに設定します．つまり，RFC5322.From のドメイン名が `example.jp` であった場合，DMARC レコードの取得は，`_dmarc.example.jp` ドメイン名の TXT レコードに対する参照となります．また，対象ドメイン名（RFC5322.From のドメイン名）である RFC5322.From に DMARC レコードが設定されていなかった場合，RFC5322.From の組織ドメイン名に対して DMARC レコードを取得します．組織ドメインについては，1.5.2 で解説します．

DKIM および SPF の認証処理は，これまで通りそれぞれの手順に従い認証処理を実施します．

識別子アラインメント (Identifier Alignment) とは，SPF あるいは DKIM で認証されたドメイン名と，RFC5322.From のドメイン名が一致した識別子（ドメイン名）のことです．ドメイン名を一致させる方法は，SPF と DKIM での識別子アラインメントモード (Identifier Alignment mode) によって異なります．

DMARC の認証の特徴には，SPF か DKIM のどちらか一方で認証が成功し，そのドメイン名が DMARC の認証ドメイン名と一致すれば，DMARC 認証は成功 (pass) となります．例えば，メール転送などの再配送により SPF 認証が失敗するなどの問題が生じた場合でも，DKIM 認証を同時に行うことにより，より安定的に DMARC 認証を運用することが可能となります．

DMARC 認証が失敗した場合，DMARC レコードで宣言しているポリシー（p=で指定）に基づいた受信処理をします．

このように DMARC では，メール受信者が参照することができる一般的な送信者情報である，From: ヘッダのドメイン名を認証するので，メール受信者にとってもわかりやすい認証技術と言えます．

1.5.2 組織ドメイン

DMARC では，メールに利用するドメイン名全てに，個別に DMARC レコードを設定しなくても良い仕組みがあります．DMARC の認証をする受信側は，RFC5322.From のドメイン名に DMARC レコードが設定されていない場合，その組織ドメインとよばれる上位ドメイン名の DMARC レコードを取得します．例えば RFC5322.From のドメイン名が mail.example.jp であり，_dmarc.mail.example.jp に DMARC レコードが無かった場合，_dmarc.example.jp に対して DMARC レコードを取得しようと試みます．DMARC レコードが取得できた場合，その DMARC レコードのポリシー等が当該メールにも適用されます．この仕組みは，SPF あるいは DKIM それぞれで，アラインメントモードが relaxed mode の場合に適用されます．

組織ドメイン名の求め方は，明確なルールや手順を決めているわけではなく，経験則に基づく手法（ヒューリスティック）によって推定されます．具体的には，TLD[*5]や日本 (.jp) の属性型 JP ドメイン名 (.co.jp) のように，通常取得できないドメイン名 + ラベル（1 つ）とされています．こうした通常取得できない上位ドメイン名は，パブリックサフィックスリスト[*6]として管理されています．

1.5.3 送信側の設定

メール送信側では，メール受信側で DMARC 認証ができるように，SPF や DKIM を導入します．確実に DMARC として認証できるようにするためには，両方を導入することが推奨されています．送信側として，SPF あるいは DKIM が導入されていれば，送信側で DMARC の導入に必要な作業は対象のドメイン名に対して DMARC レコードを設定することです．DMARC レコードは，DNS 上のテキスト (TXT) 資源レコードに設定します．設定するドメイン名は，_dmarc サブドメイン名です．例えば，example.jp ドメイン名に DMARC レコードを登録する場合は，以下のような設定となります．

DMARC レコードの設定例

```
_dmarc.example.jp.  IN TXT "v=DMARC1; p=none; rua=mailto:report@example.jp"
```

[*5] Top Level Domain: com, net, jp など

[*6] Public Suffix List: https://publicsuffix.org

DMARC レコードの記述内容や設定できるパラメータについては，1.5.4 で解説します．

DMARC では，メールに利用する RFC5322.From のドメイン名以外に，組織ドメイン名に対して DMARC レコードを設定することができます．組織ドメイン名については既に説明しましたが，そこに DMARC レコードを設定する利点は，その組織ドメイン名の配下のドメイン名全てに同じポリシーを適用できることです．もちろん，特定のサブドメイン名に対して固有のポリシーを設定する場合には，そのサブドメイン名に対してのみ，DMARC レコードを設定することになります．

組織ドメインと DMARC レコード

```
_dmarc.mail.example.jp.   IN TXT  "v=DMARC1; p=reject"
_dmarc.example.jp.        IN TXT  "v=DMARC1; p=none"
```

例えば上記の例では，RFC5322.From のドメイン名が，magazine.example.jp である場合，magazine.example.jp に対する DMARC レコードは設定されていないので，その組織ドメイン名である example.jp の DMARC ポリシーである（p=none）が適用されます．一方で，mail.example.jp ドメイン名の DMARC レコードは宣言されていますので，その DMARC ポリシー p=reject が適用されます．逆に，組織ドメイン名にのみ特定の DMARC ポリシーを設定し，サブドメイン名には適用させない，という方法もあります．これは，SPF および DKIM それぞれの認証手法毎に指定可能で，strict モードとして設定すれば，その組織ドメイン名に対してのみ認証結果が適用されます．指定しない場合は，サブドメイン名に対しても適用する relaxed モードとして解釈されます．

1.5.4 DMARC レコード

DMARC レコードは，DNS 上の TXT 資源レコードに設定します．TXT 資源レコードは，様々な用途に利用される汎用的なレコードです．そのため，他の設定と区別するために TXT レコードに記載されている文字列の先頭が v=DMARC1 と設定されている場合にのみ，DMARC レコードとして扱うことになっています．DMARC レコードの形式は，DKIM レコードと同様に，「タグ名=タグ値」のタグ形式で複数のパラメータ（タグ）が設定できるようになっており，それぞれのパラメータはセミコロン（;）で区切ります．DMARC レコードで利用できるタグを，表 1.7 に示します．

SPF および DKIM の識別子アラインメントのモード (aspf，adkim) とは，それぞれで認証したドメイン名と DMARC の認証ドメイン名 (RFC5322.From) との関係を示す設定です．relaxed mode (r) の場合，SPF あるいは DKIM での認証ドメイン名と RFC5322.From ドメイン名の組織ドメイン名が同じであれば，認証したドメイン名，つまり識別子 (identifier) が Identifier Alignment（認証ドメイン名）となります．例えば，SPF の認証ドメイン名が cbg.bounces.example.com であり，RFC5322.From が payments@example.com であった場合，SPF の識別子アラインメントのモードが relaxed mode(r) の場合はそれぞれが同列 (in alignment) となり，Identifier Alignment となります．strict mode(s) では，FQDN が一致している必要がありますので，上記の例では認証が失敗します．

これらのタグの中で，必ず設定しなければならない必須のものは，v タグと p タグです．DMARC レコードは，その重要な目的であるポリシー（p=reject）を設定することから，DMARC ポリシーレコードとも呼

表 1.7　DMARC レコードのタグ

タグ名	タグ値	説明	省略の可否
v	バージョン番号	・レコードの最初に DMARC1 とする	必須
p	受信側に要求するポリシー	・受信側に要求するポリシー ・sp タグで明示的に示さない限りサブドメイン名にも適用される ・設定可能な値は以下 none 特別な処理はしない quarantine 認証失敗時に不審メールとして扱うことを求める reject 認証失敗時に受け取り拒否することを求める	必須
sp	サブドメイン名に対するポリシー	・受信側に要求する全サブドメイン名に対するポリシー ・設定可能な値は p タグと同じ ・対象はサブドメイン名だけで当該ドメイン名に対するものではない ・省略時は p タグと同じポリシーとなる	省略可
rua	集約レポートの宛先	・集約レポートの宛先を URI（mailto: とメールアドレス）で示す ・宛先はカンマ (,) で区切り複数記述可能 ・宛先が当該ドメイン名と異なる場合には別途設定が必要	省略可
ruf	失敗レポートの宛先	・失敗レポートの宛先を URI（mailto: とメールアドレス）で示す ・宛先はカンマ (,) で区切り複数記述可能 ・宛先が当該ドメイン名と異なる場合には別途設定が必要	省略可
pct	ポリシー適用割合	・段階的な導入を促すためのポリシーを適用すべきメールの割合 ・設定可能な値は 0〜100 の数値（省略時は 100 と判断）	省略可
adkim	DKIM の識別子アラインメント	・DKIM の識別子アラインメントのモード ・設定可能な値は以下（省略時は r） r(relaxed mode) s(strict mode)	省略可
aspf	SPF の識別子アラインメント	・SPF の識別子アラインメントのモード ・設定可能な値は adkim と同じ (r, s)	省略可
ri	集約レポートの要求間隔	・メール受信側に要求する送信間隔 ・秒単位で指定（指定無しは 86400） ・但し 1 回/日は送信されなければならない	省略可
rf	失敗レポート形式	・レポートの形式を指定（指定無しは afrf）	省略可
fo	失敗レポートの要求基準	・レポートを送信する条件を指定（省略時は 0） 0: 全ての認証結果が pass でない 1: いずれかの認証結果が pass でない d: DKIM の認証が失敗 s: SPF の認証が失敗	省略可

ばれます.

pct は，ドメイン管理者が設定した DMARC レコードのポリシー（p または sp）を受信側が適用する割合を 0...100 の間で示すタグです．これは，より強いポリシーの設定を躊躇してしまうことを緩和する役割をはたします．例えば DMARC のポリシーが reject に該当する場合は，メッセージを受信拒否すべきですが，pct タグにより対象とならない場合には，quarantine のポリシーを適用すべきとなっています．DMARC のポリシーが quarantine の場合は，通常の取り扱いをすることになります．ただし，DMARC レポートのデータに関しては，この pct の影響を受けないことになっています.

1.5.5 DMARC レポート

DMARC レポートは，送信ドメイン側が DMARC 認証結果を含む送信ドメイン認証の状況を把握するための仕組みです．DMARC レポートは，メール受信側からドメインの管理側にメールとして送信されます．DMARC レポートには，集約レポート (aggregate report) と失敗レポート (failure report) があります．

失敗レポートは，メール受信時に DMARC 認証の元となる SPF，DKIM が失敗し，DMARC レコードの `fo` タグで指定された条件に適合した場合に，失敗要因を示す即座に送信されるレポートです．レポートの送信先は DMARC レコードの `ruf` タグで示します．レポートの形式は，現在のところ AFRF(Authentication Failure Reporting Format, RFC5965) だけが定義されています．通知手段としてはメールを利用します．AFRF 形式の基本的なフォーマットは，失敗要因等を MIME ヘッダで示し，元のメールを MIME データとして取り込んだものです．

集約レポートは，一定間隔で DMARC および SPF, DKIM の認証結果とそれに基づく受信側の処理結果を示すレポートです．レポートの送信先は DMARC レコードの `rua` タグで示します．原則として DMARC レコードの `ri` タグで示された送信間隔で送信されます．集約レポートに含まれる情報としては，メールの送信元（IP アドレス）毎に，SPF，DKIM それぞれの認証結果，SPF，DKIM それぞれの識別子がアラインメントであるかどうかの結果，受信側で適用したポリシーの結果などが含まれます．集約レポートでは，これらの情報を XML 形式で表現し，それを gzip 形式[*7]で圧縮したファイルを MIME 形式（いわゆる添付ファイル）で送信することになっています．集約レポートのスキーマ構造を図 1.11 に示します．

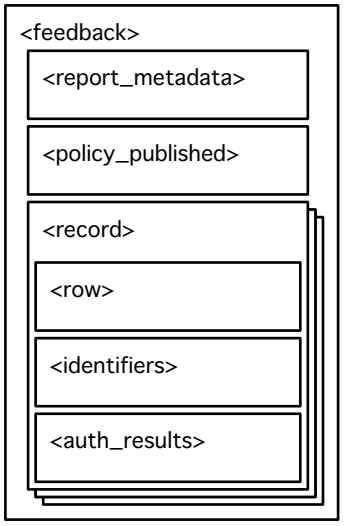

図 1.11　DMARC 集約 (aggregate) レポートの XML Schema

DMARC 集約レポートでは，`<record>`部分に送信元（IP アドレス）毎に受信したメール数，それぞれの認証技術での認証ドメイン名，認証結果などの情報が示されます．

[*7] 実際には zip で送られる場合もあるようです

DMARC レポート受信の委譲

ドメイン管理者は，DMARC レポートの宛先として自ドメイン名以外の宛先を DMARC レコードに設定することができます．しかし，DMARC レポートの送信機能を悪用し，無関係の第三者に不要な DMARC レポートが大量に送信されてしまう可能性があります．こうした事態を防ぐために，DMARC レポートの送信者，つまり DMARC を認証し結果をレポートとして送信するメールの受信側は，DMARC レポートの送信先の確認が必要となっています．具体的には，宛先のドメイン名が認証したドメイン名と異なる場合，委譲関係があるかを調べます．委譲関係は，図 1.12 のように，委譲先の DMARC レコード（内容は DMARC のバージョン番号）を委譲元のドメイン名を利用したサブドメイン名に設定することで示します．

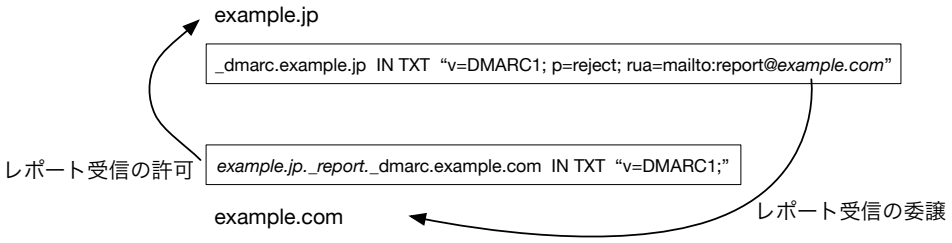

図 1.12　DMARC 集約 (aggregate) レポートの委譲の関係

例えば DMARC レポートを代わりに受け取り，内容を解析して情報提供するようなサービスでは，多くのドメイン名のレポートを受け取る必要があります．このようなサービスでは，`_report._dmarc` サブドメイン名配下に対して，ワイルドカードで DMARC レコードを設定するなどの対応が考えられます．

1.5.6　受信側の設定

メール受信側で DMARC 認証を行うためには，SPF と DKIM 両方の認証機能を組み込む必要があります．SPF と DKIM とで認証した結果を用いて，DMARC としての認証を行うためです．そのためメールの受信側では，SPF と DKIM の認証機能を組み込んだ上で新たに DMARC の認証機能を組み込むことになります．

DMARC 認証が失敗したメールに対しては，DMARC ポリシーに基づいた処理を行うことができます．ポリシーが `quarantine` の場合には，「迷惑メールフォルダ」等の通常の受信メール以外の場所に保存します．ポリシーが `reject` の場合は，受信を拒否します．いずれの場合でも，DMARC に準拠したメール受信処理を実施していると明確に示していない限り，受信したメールの取り扱いについては受信側で判断することになります．また，メールの送信ドメインの管理者が DMARC ポリシーを設定したとしても，実際のメール受信者がそうした受信処理を望まない可能性があります．DMARC ポリシーに基づいた受信処理を行う場合には，メール受信者に十分な周知が必要です．なお，電気通信事業者等は，送信ドメイン認証結果によるラベリングやその結果に基づくフィルタリングの実施，DMARC レポートを送信する場合には，事前に利用者に対して十分な説明を行うとともに契約約款等で利用者の同意を取得すること等により実施が可能となっています．認証結果のラベリングとは，送信ドメイン認証の結果を 1.6 で示すようなメールヘッダに保存することです．

送信するメールが SPF，DKIM，DMARC 全て対応している場合でも，正規のメールが DMARC 認証に失敗することがあります．SPF や DKIM が認証失敗する例として，2.4 でメール再配送の課題を解説しています．受信メールの取り扱いは，メールの利用形態や送信側の状況に応じて，総合的に判断すべきです．

1.6 認証結果のヘッダへの表示

送信ドメイン認証技術による認証結果は，受信したメールに対して Authentication-Results ヘッダに記録されます．Authentication-Results がヘッダ領域名 (field name) となり，コロン (:) に続いてヘッダ本体 (field body) が続き，認証 ID や認証結果，関連情報などが記載されます．メールヘッダの書式は，RFC5322 で規格化されており，Authentication-Results ヘッダも基本的にはこの書式に従います．メールヘッダが長くなる場合は，折り返して先頭行をタブや空白から開始することで継続行であることを示します．Authentication-Results ヘッダの正確な書式は，RFC8601 で規格化されています．以下にヘッダ例を示します．

Authentication-Results ヘッダの例

```
Authentication-Results: example.jp;
    spf=pass smtp.mailfrom=example.net;
    dkim=pass (good signature) header.i=@example.net;
    dmarc=pass header.from=bob@example.net
```

ヘッダ領域名に続くホスト名 (上記では example.jp) 部分は，認証サービス識別子 (authserv-id) と呼ばれます．この認証サービス識別子は，示される認証結果が誰によって認証されたのかを示す情報なので，メール受信者や MUA などにとって重要な情報となります．そのため，送信ドメイン認証を行う MTA では，同じ認証サービス識別子が付けられた Authentication-Results ヘッダが既に存在している場合は，そのヘッダを削除するべきとしています．これは認証結果の偽造を防ぐためにも重要な確認処理です．

Authentication-Results ヘッダには，複数の認証機構の結果を示すことができます．各認証結果は，セミコロン（;）で区切られます．上記の例では，SPF, DKIM, DMARC による認証結果が一つの Authentication-Results ヘッダに示されています．

1.6.1 認証機構と結果

Authentication-Results ヘッダに記述できる認証機構とそれによる結果の値は，IANA (Internet Assigned Numbers Authority) によって登録[*8] されます．認証機構は method = result の形式で示されます．認証結果は，理由を示す文字列や 1 つ以上の property = value の形式が含まれます．送信ドメイン認証技術に関連する認証機構と認証結果の組み合わせを表 1.8 に示します．

送信ドメイン認証技術以外の認証機構としては，auth (SMTP-AUTH) や iprev (IP アドレスの逆引きと正引きの評価) などもあります．

[*8] https://www.iana.org/assignments/email-auth/email-auth.xhtml

表 1.8　送信ドメイン認証機構と認証結果

認証機構	認証結果	意味
SPF	pass	認証されたことの明示的な提示
	fail	認証できなかったことの明示的な提示
	softfail	おそらく認証できなかったことを示す弱い提示
	neutral	認証できたかどうかを明らかにしない
	none	認証できる識別子がなかったか SPF レコードがなかった
	permerror	SPF レコードを正しく解釈できなかった
	temperror	DNS などの要因で取得できなかった
	policy	認証されたがローカルなポリシーによって受け入れられない
DKIM	pass	署名があり，検証が通り受け入れられる
	fail	署名があるが検証が通らない
	neutral	署名があるが内容に構文エラーがあるなどの理由で処理が行えない
	none	署名がない
	permerror	必要なヘッダが含まれていないなど検証処理が行えなかった
	temperror	公開鍵が取得できなかったなどにより検証できなかった
	policy	署名があるが何らかの理由により受け入れられなかった
DMARC	pass	DMARC レコードがあり一つ以上の認証機構がパスした
	fail	DMARC レコードがありパスした認証機構が無かった
	permerror	DMARC レコードの書式が違うなど永続的なエラー
	temperror	DMARC 評価中の一時的なエラー
	none	DMARC レコードが公開されていないか識別子が取得できない

1.6.2　認証機構とプロパティ

送信ドメイン認証技術では，それぞれで認証するプロパティがあります．プロパティは，プロパティタイプ (`ptype`) とプロパティ (`property`) を'.' で結合します．これをプロパティの値 (`pvalue`) を '=' で結合します (`ptype.property = pvalue`)．それぞれの認証機構で利用できるプロパティを表 1.9 に示します．

表 1.9　送信ドメイン認証機構とプロパティ

認証機構	ptype	property	意味
SPF	smtp	mailfrom	エンベロープ上の送信者 (認証対象外を除く)
	smtp	helo	HELO/EHLO の値
DKIM	header	d	署名の `d` タグの値
	header	i	署名の `i` タグの値
	header	b	署名の `b` タグの値
	header	a	署名の `a` タグの値
	header	s	署名の `s` タグの値
DMARC	header	from	RFC5322.From のドメイン部分
	policy	dmarc	`pct` や `sp` などのポリシーオプション適用後の DMARC 評価

第 2 章

応用編

2.1　送信ドメイン認証技術導入手順

送信ドメイン認証技術は，メールの送信側と受信側の双方が協調することによって成り立っています．送信ドメイン認証技術は，既存のメール配送の仕組み (SMTP) に直接影響を与えることなくメールの送信側と受信側のどちらの立場からも導入することができます．

第 1 章の基礎編において，送信ドメイン認証技術 SPF, DKIM, DMARC それぞれの基本的な仕組みや設定方法については解説しました．応用編では，送信ドメイン認証技術の導入に際し，DMARC 認証が有効に機能する観点から，事前に検討しておくべきことや，送信側と受信側それぞれの側での作業にあたり留意すべき点，導入後の運用にあたり注意すべき点などについて解説します．

また近年では，メール機能の一部あるいは全体を，クラウド等を利用した外部の事業者が提供するサービスを利用したり，メールマガジンなど特定目的のメール配信を送信事業者に外部委託するなど，他の事業者が提供するサービスと連携した利用が進んでいます．外部の事業者との連携する方法や，いわゆるクラウドサービスを利用した場合でも，正しく送信ドメイン認証技術が利用できる設定運用方法についても解説します．

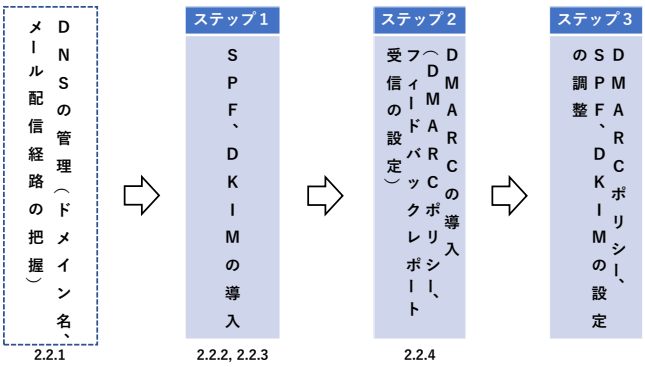

図 2.1　DMARC の導入手順

2.2 メール送信側の設定

本節では，主にメール送信側として設定する内容について解説します．設定する内容としては，SPF の場合は SPF レコードの設定，DKIM の場合は送信メールサーバへの DKIM 署名機能の導入と DKIM レコードの設定，DMARC の場合は DMARC レコードの設定がそれぞれ必要となります．また，DMARC レポートを受け取る場合には，そのための設定も必要となります．

それぞれの送信ドメイン認証技術や関連技術ごとに，設定に関して注意すべき点や有益となる設定方法について解説します．また，SPF, DKIM は，それぞれ独立した送信ドメイン認証技術ですが，ここでは最終的に DMARC で認証できるための設定方法を中心に解説します．

2.2.1 DNS の管理

送信ドメイン認証技術は，DNS に大きく依存した形で実現されています．SPF, DKIM, DMARC を送信側として導入するためには，DNS に対して対象となる送信ドメイン名に関する情報を正しく設定する必要があります．また，組織内でメールの管理運用の担当者と，DNS の管理運用の担当が異なっている場合や，DNS の設定を外部委託している場合など，双方での連携や意思疎通など，適切な情報共有が必要となります．

自組織のドメイン名を DNS に設定し運用する方法として，外部事業者（DNS サービス事業者）に委託する場合が多いと思います．DNS サービス事業者によっては，送信ドメイン認証技術の送信側の設定をする上で，必要となる設定ができるのかを予め確認しておく必要があります．具体的には，以下の項目です．

- ドメイン名のラベルとして “_”（アンダースコア）が先頭に利用できること
- テキスト資源レコード (TXT RR) を自由に設定できること
- テキスト資源レコードが十分な長さで記述できること
- CNAME 資源レコードの設定ができること（DKIM 署名を外部の事業者が行う場合等）

DNS で設定するラベルの先頭にアンダースコアを利用する場合としては，DKIM レコード，DMARC レコードの設定に必要となります．CNAME 資源レコードは，いわゆる別名定義に用いられる資源レコードですが，クラウド型メールサービスでの DKIM 署名に利用する鍵を，利用するサービス側（クラウド側）で管理する場合に利用することがあります．そのため，特にテキスト資源レコードに対する CNAME 資源レコードを設定できることが必要になる場合があります．詳細は，2.2.3 項で解説します．

さらに，こうした DNS に設定される情報の信頼性を高めるためには，DNSSEC(DNS Security Extensions) に対応しておくことも有効です．利用する DNS サービス事業者および管理しているドメイン名が，DNSSEC に対応しているのかを確認することも必要です．

DNS サービス事業者を利用した DNS の設定には，Web 上で各種の操作を行うことが多いと思います．これらの設定の参照や変更を行うには，Web 上で認証用の ID とパスワードを入力するログイン作業が一般に必要となります．このログインのための認証情報が外部に漏れると，送信ドメイン認証関連の設定はもちろん，メールの配送先を示す MX 資源レコード等の設定も変更されてしまうおそれがあります．DNS サービスに限りませんが，Web 上で操作する各種サービスを利用する場合は，認証のための情報（認証 ID やパスワード）が無関係の第三者に漏れないように管理すべきです．また，サービス提供側も，ログイン時に多要素認証を必要とするなど，不正なログインを防ぐ仕組みを導入するべきです．

2.2.2 SPF レコードの設定

SPF レコードには，メール受信側からみてメールを送信するホスト情報（IP アドレス）を記述します．メール送信ホストに変更がなければ，基本的には SPF レコードを変更する必要は無いので，定期的な作業は必要ありません．

■外部サービスの利用 メールサービスの提供元は，提供先で SPF レコードの設定を容易にするため，メールサービスで利用する送信元のメールサーバの IP アドレスを示した SPF レコードを提供すべきです．メールサービス利用元は，この提供された SPF レコードを include 機構を利用して IP アドレスを取り込みます．このような取り込み (include) 用の SPF レコードを用意し利用してもらうことで，メールサービス提供元がメールサーバの構成を変更する場合や，IP アドレスを書き換えた場合でも，利用元にその都度連絡することなく変更を反映させることができます．また利用元でも，自ドメイン名の SPF レコードを書き換えることなく構成変更に対応できます．

以下に include の利用例を示します．メールサービスの提供元の SPF レコードが mail.example.net であり，メールサービスの利用元のドメイン名を example.jp とした場合の例を以下に示します．

include の利用例

```
mail.example.net.   IN   TXT   "v=spf1 ip4:203.0.113.1/26 -all"
example.jp.         IN   TXT   "v=spf1 include:mail.example.net -all"
```

■複数ドメインの管理 様々な経緯や目的により，メールに利用するドメイン名を複数管理する場合があります．例えば，部門毎に異なったドメイン名を利用したいと考えたり，何らかのキャンペーン等で配信するメール用にドメイン名を変えたい場合などです．これら複数のドメイン名が実際には同じメールシステムで利用する場合には，SPF レコードの redirect 修飾子が利用できます．

redirect の利用例

```
tokyo.example.jp.   IN   TXT   "v=spf1 redirect=spf.example.jp"
osaka.example.jp.   IN   TXT   "v=spf1 redirect=spf.example.jp"
spf.example.jp.     IN   TXT   "v=spf1 ip4:203.0.113.1/26 -all"
```

上記の例では，tokyo.example.jp や osaka.example.jp の SPF レコードは，redirect 修飾子により spf.example.jp の SPF レコードに向け直されます．これにより，メールシステムに変更があった場合でも，spf.example.jp ドメイン名の SPF レコードを修正するだけで，他の 2 つのドメイン名の SPF レコードを修正することなく反映させることができます．

include 機構との違いについても触れておきます．include 先の SPF レコードで pass の結果にならなかった場合，SPF 認証の失敗の強度は include 先の失敗の強度にはならず，あくまで include 機構部分が適合しないだけであり，その後に記述された all の限定子によって決まります．redirect 修飾子の場合は，

redirect 先の認証の結果が元のドメイン名の認証結果になります．そのため，redirect 修飾子は SPF レコードの末尾に記述するべきであるとされ，all 機構がある場合は，redirect 修飾子は無視しなければならないとなっています．

■DNS 参照回数の上限　SPF レコードを設定する上で注意すべき点は，SPF 認証するメール受信側で，認証処理中に DNS の参照回数が 10 回を超えないように SPF レコードを記述することです．参照回数が 10 回を超えた場合，認証結果としては permerror となり pass しない（認証できない）ことになります．この仕組みは，DNS への問い合わせを多く行うことで，DNS に過度な負荷を与えてしまうことを防止する目的で制限が設けられています．

こうした制約のために SPF レコードには，メールの出口を示す機構として ip4 や ip6 など IP アドレスを直接記述することが望ましいと言えます．SPF レコードで利用可能な機構で，ドメイン名を指定する a や MX レコードを示す mx は，DNS の参照回数が増える記述方法です．機構 a では，指定されたドメイン名から IP アドレスを取得するため，DNS 参照回数が 1 回増えます．機構 mx では，MX レコードに指定されたドメイン名の取得と，そのドメイン名から IP アドレスを取得するための DNS 参照で 2 回増えることになります．

DNS 参照が増える SPF レコード例

```
mx.example.jp.     IN  A     192.0.2.1
mail.example.jp.   IN  MX    mx.example.jp
mail.example.jp.   IN  TXT   "v=spf1 mx -all"
...
msg.example.jp.    IN  TXT   "v=spf1 a:mailhost.example.jp -all"
```

外部のメールサービスや配信サービスを利用する場合に利用する include 機構を利用すると，include 先の SPF レコードを取得するための DNS 参照が 1 回行われます．include 先の SPF レコードがさらに include 機構を利用していたり，ホスト名などを設定している場合には DNS の参照回数が増えます．そのため，あらかじめメールサービス提供側の SPF レコードでも，なるべく DNS の参照回数が発生しないような記述をすべきです．また，SPF レコード設定時点で参照回数が 10 回以下だったとしても，include 先の SPF レコードの管理元の設定変更により DNS の参照回数が増えるような変更をした場合に，include 元で SPF レコードを変更していなくても参照回数が 10 回を超えることもありますので，注意が必要です．

こうした観点から SPF レコードの設定には，以下の事柄に注意するべきです．

- 設定する送信元は IP アドレスあるいはネットワークアドレス（CIDR）を用いてなるべく少ない文字数で記述指定する
- ネットワークアドレスには必ずしもメールサーバに使われるアドレスだけを記述する必要は無く，自組織管理の範囲内であればネットワークアドレスとしてまとめることで記述項目を減らす
- a，mx，ptr，exists などの機構はその部分自体で DNS 参照を伴うので利用時に注意が必要（全体で DNS 参照回数を超えないように）
- 特に機構 mx は，MX レコードの参照とそのドメイン名に対する IP アドレス（A または AAAA レコード）の参照を伴うため最低 2 回の DNS 参照を必要とする（MX レコードには直接 IP アドレスが設定

できない）

SPF レコードでのネットワークアドレスによる指定方法は，送信サーバが複数存在し，それらが送信メールサーバ以外の用途のホストを含む特定のネットワークアドレス内に収まっているのであれば，ネットワークアドレスで送信サーバを指定することで，SPF レコードの記述量を減らすことができます．指定したネットワークアドレスに，送信メールサーバが含まれていなくても，それらの IP アドレス帯域を所有管理しており，メールが送信される可能性が無いのであれば，それらをネットワークアドレスで指定しても問題ありません．

SPF レコードのネットワークアドレス指定

```
mailout1.example.jp.   IN  A    192.0.2.4
mailout2.example.jp.   IN  A    192.0.2.6
example.jp.            IN  TXT  "v=spf1 192.0.2.4/30 -all"
```

メールの利用形態は様々あり，今後の用途の広がり（外部サービスの利用等）も考慮するのであれば，SPF レコードを設定する時点で各種制限（DNS の参照回数や記述文字列数について）に近い設定をするのではなく，余裕を持たせておくことが望ましいといえます．SPF レコードの設定については，運用上の手間を省略したい目的でホスト名を指定する場合もありますので，SPF レコードの設定内容は，全体的な運用負荷と SPF 認証時の DNS 参照回数制限の両方を検討したうえで総合的に判断すべきです．

■SPF レコードの設定エラー　正しく SPF レコードを設定し，最初は SPF 認証できていたとしても，何らかの事情で include や redirect で指定したドメイン名の SPF レコードが無くなっている場合があります．これら指定したドメイン名の SPF レコードが参照できない場合[*1]，SPF としては permerror となります．自組織の管理外の SPF レコード（ドメイン名）を参照先として指定している場合には，その SPF レコードが存在しているか，あるいはメール受信側で SPF の認証結果が permerror となっていないかなど，定期的に確認すべきです．

SPF の認証結果が permerror となってしまう設定例に，同じドメイン名に複数の SPF レコードを設定している場合があります．SPF では，SPF レコードとして TXT 資源レコードを利用するため，複数の SPF(TXT) レコードを DNS の仕様上は設定可能です．しかし，SPF の仕様として同じドメイン名に複数の SPF レコードが存在する場合は permerror となります．

複数の SPF レコードの設定例 (permerror)

```
example.jp.   IN  TXT  "v=spf1 ip4:203.0.113.1/26 -all"
example.jp.   IN  TXT  "v=spf1 ip6:2001:db8::/32 -all"
example.jp.   IN  TXT  "v=spf1 include:mail.example.net -all"
```

上記の例のように，複数のメール送信サーバが存在する場合は，以下のように 1 つの SPF レコードにそれ

[*1] DNS の応答として NXDOMAIN エラーだった場合など

ぞれのメールサーバの情報を記述します.

送信メールサーバの送信経路が複数の設定例

```
example.jp.   IN   TXT   "v=spf1 ip4:203.0.113.1/26 ip6:2001:db8::/32 include:mail.example.net -all"
```

SPF レコードの記述間違いによって，permerror となる事例として記述間違いがあります．良く見られる記述間違いを以下に示します.

- 区切り文字としてセミコロン（;）をつけたり区切り (空白) の記述漏れ，SPF では各項目の区切りは一つ以上の空白文字です
- IP アドレスを示す機構として ipv4 や ipv6 と v を付ける，正しくは ip4，ip6 です
- 限定子と機構の間に空白を入れてしまう（ex. +␣mx）

これら SPF レコードの設定に間違いが無いかを確認するためには，チェックサイトを利用したり，SPF 認証を実施している受け取ることができる宛先に，実際にメールを送信するなどで確認することができます.

■DMARC 認証のための SPF 認証　SPF では，正規のメール送信元以外を対象とするために all を末尾に置いて，その限定子 (qualifier) を −，∼，?の中から選ぶことで，認証失敗の強度を fail，softfail，neutral と変化させることができます．しかしながら DMARC の観点からは，SPF 認証が成功したドメイン名しか扱いませんので，これらの認証失敗の強度の違いは，DMARC としては意味がありません．もちろん，SPF としての認証結果としては意味はありますし，その結果を利用した受信処理を行う可能性があるので，送信側として適切な限定子を設定すべきです.

DMARC 認証のための SPF 認証として気をつけなければならないことは，SPF 認証されるドメイン名 (RFC5321.From あるいは HELO/EHLO のドメイン）と DMARC 認証のドメイン（RFC5322.From のドメイン）との関係です．通常のメール送信において，一般的なメールソフトウェア (MUA) からメール送信する場合，エンベロープ From(RFC5321.From) とヘッダ From(RFC5322.From) の送信ドメインが異なることは，通常ありません．しかしながら，以下のようなメールの利用形態では，それぞれの送信ドメインが一致しないことがあります.

- 何らかのキャンペーンメールやメールマガジン等の送信を外部の送信事業者に配信を委託する場合
- 外部のドメイン名で運営されているメーリングリスト経由で配送される場合
- 共有型のメールサービスの利用時に RFC5321.From が空 (<>) を指定したメール送信の場合（共有型サービスでの独自ドメイン等が HELO/EHLO のドメインとなる）

こうした利用形態では，SPF で認証 (pass) できたとしても DMARC では認証できない (fail) ことがあります．このような場合，DKIM 署名を併用することで解決できる場合があります．また，DKIM を利用した場合でも注意すべき利用形態がありますので，通常のメールの使い方をしていたとしても，メール送信側だけではうまく DMARC 認証できない場合があります．こうした，メール再配送 (indirect email flow) の課題と対策については，**2.4 メール再配送の課題**で述べます.

2.2.3 DKIM の設定

DKIM の公開鍵は，DKIM-Signature ヘッダに示された d，s タグの情報を利用し，参照するドメイン名のテキスト資源レコード (TXT RR) に設定します．そのため，署名時の処理（利用する秘密鍵）と DNS に設定されている DKIM の公開鍵は必ず連動（ペア）しなければなりません．少なくとも，メール受信側にメールが届き DKIM 署名を検証する時点で，DNS から DKIM の公開鍵が正しく取得できるよう予め設定しておく必要があります．

■DKIM の鍵管理　DKIM レコードに設定する公開鍵とペアとなる秘密鍵は，DKIM 署名の作成時に必要となりますので，外部に漏れないよう適切に管理する必要があります．DKIM 署名の秘密鍵が漏れると，ドメイン名の管理元と無関係な第三者が，当該ドメイン名で認証できる DKIM 署名が作成可能となり，DKIM 認証ができる詐称メールが送信できてしまいます．また，DKIM 認証ができれば DMARC 認証できるメールも送信できることになります．このため，DKIM 署名の秘密鍵を漏洩しないように管理するとともに，万が一漏れてしまったり類推できる状態となった場合には，速やかに新しい鍵ペアに交換すべきです．そのため新しい鍵の生成や鍵交換のための手順と運用体制を準備しておくべきです．

メール送信の一部を外部に委託するような場合，できれば利用する鍵はそれぞれで管理運用すべきです．そのためには，サブドメイン名を利用するなどそれぞれで別ドメイン名とするか，セレクタ名を変えるなどの手法を用います．これらの手法では，公開鍵情報を含む DKIM レコードの位置が変わりますので，それぞれ別の鍵情報を利用できます．DKIM レコードの設定については，DNS 設定をする側が公開鍵の情報を受け取ったり，ネームサーバの委譲や別名定義をするなどの方法があります．

以下にメールの送信を別組織で行う場合の手法を示します．

- 送信委託先で鍵ペアを作成，公開鍵を入手し DNS に DKIM レコードを設定する（別セレクタ，サブドメインの利用）
- 鍵ペアを作成し，安全な方法での秘密鍵を委託先に渡し管理運用してもらう（別セレクタ，サブドメインの利用）
- サブドメインを作成し，委託先に管理を委譲する（但し，DKIM 以外のレコードも作成される可能性があるので注意が必要）

メール送信委託先が利用する鍵ペアを別にすることで，秘密鍵が漏れた場合の責任や影響をある程度限定させることができます．

■署名鍵の交換方法　DKIM 署名に用いる秘密鍵は，DNS の DKIM レコードに示される公開鍵とペアになっています．これらの鍵ペアは，鍵の安全性の観点から定期的に変更することが望ましいとされています．鍵ペアを変更する場合，受信側が認証する際に参照する，DKIM レコードに設定されている公開鍵を参照するタイミングも考慮しなければなりません．これらのことから，一定の期間の間，新旧の鍵ペアを共存させておく必要があります．

新しい DKIM レコードがメール受信側で参照できるようになるまでの時間（TTL 等）も考慮しておく必要があります．DKIM 署名に新しい秘密鍵を使う場合は，十分な時間的余裕をもって，対応する DKIM レコードを設定しておく必要があります．

DKIM では，セレクタを複数用意しておくことで，複数の鍵を共存させることができます．一般的な DKIM

の署名鍵の交換手順について示します.

1. 新しい鍵ペアを作成する
2. 新しいセレクタ名を作成し，鍵ペアの公開鍵を含む新しい DKIM レコードを新しいセレクタ名に設定する
3. 十分な時間の経過ののち，新しい鍵ペアの秘密鍵を DKIM 署名に利用する．新しいセレクタ名を DKIM-Signature ヘッダに記述する
4. さらに十分な時間の経過後，古いセレクタ名の DKIM レコードを削除する

■署名機能の外部利用　メール送信の一部を外部委託する場合の DKIM 署名のための鍵の管理については既に述べました．ここでは，外部委託先に DKIM レコードを委譲するもう一つの方法，CNAME 資源レコードを利用する方法について述べます.

CNAME は，正式名称に別名を付けるために利用される資源レコードで，これを DKIM の鍵管理および DKIM 署名を行うドメイン名を示す別名として設定すれば，あたかも自分のドメイン名で DKIM 署名および鍵管理を行っているようにみせることができます．例えば，鍵の管理および署名を行っているドメイン名を example.com とし，example.jp ドメイン名が鍵の管理及び DKIM 署名を委譲しているとします．この場合，example.jp ドメイン名の本来の DKIM レコードが設定されるドメイン名 (key1._domainkey.example.jp) に対して，DKIM レコードを設定する代わりに，委譲先の DKIM レコードを CNAME で別名定義します.

DKIM レコードの委譲

```
key1._domainkey.example.jp.   IN   CNAME   key1.example.com
...
key1.example.com.             IN   TXT     "v=DKIM1; p=ADfe34556..."
```

この設定により，メール受信側で example.jp ドメイン名の DKIM レコードを取得するために，key1._domainkey.example.jp の問い合わせを行った場合，結果として key1.example.com に設定された DKIM レコードが得られることになります．このような設定により，DKIM として認証された場合，DKIM 認証ドメインが example.jp ドメインとなります．これにより，DMARC 認証時にもヘッダ上の送信ドメイン名と一致させることができるようになります.

実際には，既に述べた通り DKIM の鍵交換のために複数のセレクタによる複数の鍵の共存期間が必要となりますので，CNAME で委譲する DKIM レコードも複数必要となります．通常は，以下に示すように，最低限 2 つの DKIM レコードが同時にアクセスできるようになっている必要があります.

鍵交換のための DKIM レコードの委譲

```
key1._domainkey.example.jp.   IN   CNAME   key1.example.com
key2._domainkey.example.jp.   IN   CNAME   key2.example.com
...
key1.example.com.             IN   TXT     "v=DKIM1; p=ADfe34556..."
key2.example.com.             IN   TXT     "v=DKIM1; p=A783Fg4556..."
```

DKIM レコードを委譲される側は，DKIM 署名および検証に必要な鍵ペアを独自に生成管理することができます．また，委譲元が DKIM レコードとして CNAME を設定するドメイン名及びセレクタ名を予め把握しておく必要があります．これらのドメイン名及びセレクタ名を，送信するメールに追加する `DKIM-Signature` ヘッダのパラメータ (`d`，`s`) タグに設定することになります．

■鍵交換の頻度　DKIM の署名および検証に利用する鍵ペアは，安全性の観点から定期的な変更が望ましいことを示しました．では，その変更頻度はどの程度の期間が望ましいでしょうか．運用の負担からは，なるべく鍵の交換が少ない方が良いはずですが，基本的には秘密鍵が十分安全である期間はそのまま維持しても良いでしょう．安全性の観点では，秘密鍵の保管方法やそれを守っているシステムの安全性，暗号の強度（鍵の長さやアルゴリズム）によって違いがありますし，鍵を求める新しい手法やそれを実行する計算機能力の向上など，鍵の安全性の期間については，今後も変化も含めて複合的な要因で決まるパラメータであると考えます．

とはいえ，現時点で DKIM 署名を運用していくためには，何らかの指標が必要であることも事実です．グローバルなセキュリティ団体である M^3AAWG では，DKIM の鍵交換に関する BCP[*2](Best Common Practices) を発行し，その中で幾つかの前提を述べた上で，現実的には 2 回/年は変更すべき，と述べています．

■署名の無いメール　DKIM では，DKIM 署名（`DKIM-Signature` ヘッダ）があるメールだけが認証の対象となります．`DKIM-Signature` ヘッダは，メールヘッダとしては必須ではないため，`DKIM-Signature` ヘッダが無いメールは単に DKIM 認証ができないだけで，それ自体が不正なメールと判断することはできません．そうした DKIM 署名が無いメールは，そもそも最初から DKIM に対応していない送信元からのメールかもしれませんし，たまたまそのメールが何らかの事情により DKIM 署名がつけられなかったのかもしれません．また，当然ながら詐称メールなど第三者が勝手に送信したメールである可能性も十分に考えられます．

こうした背景から，DKIM ADSP(DomainKeys Identified Mail Author Domain Signing Practices) が仕様として作られました．DKIM ADSP は，簡単に言えば必須ヘッダである `From` ヘッダのドメイン名 (Author Domain) から得られる特定のドメイン名に対して，ADSP レコードを設定できるようにし，その ADSP レコードに，DKIM の署名状況と署名が無いメールの取り扱い (`unknown`，`all`，`discardable`) を記述できるようにした仕様です．

DKIM ADSP の仕様が作られたあとに DMARC の仕様が作られました．DMARC では DKIM ADSP と同様に，認証のために `From` ヘッダのドメイン名を利用すること，認証ができないメールの取り扱いを DNS を利用して示す機能を実現します．そのため，現在では DMARC が DKIM ADSP に代わる技術仕様として

[*2] M^3AAWG DKIM Key Rotation Best Common Practices. https://www.m3aawg.org/sites/default/files/m3aawg-dkim-key-rotation-bp-2019-03.pdf

認識され利用されつつあります.

■DMARC 認証のための DKIM 認証　DMARC では，SPF および DKIM 認証できたドメイン名を利用しますが，送信側としてどちらか一方だけでも，正しく認証できれば DMARC として認証することができます．つまり，送信側として導入がしやすい SPF だけを導入（SPF レコードを設定）しても DMARC 認証は可能になりますが，SPF 認証の仕組み上，最終的なメール受信者に届くまでの経路を最初のメール送信側で制御することはできません．そのため，メール転送時など SPF だけでは正しく DMARC 認証できない場合があります．こうしたメール転送時の SPF 認証失敗による DMARC 認証の失敗を防ぐためには，送信側での DKIM の導入（DKIM 署名）が必要です.

DKIM を導入し，DMARC レコードを設定すれば，確実に DMARC 認証できるとは言えない場合もありますが，DKIM を導入することで，格段に DMARC 認証が成功する割合を増やすことができます．これは，受信側での SPF, DKIM, DMARC によるそれぞれの認証結果の組み合わせをみても，現時点では明らかな結果です.

■他の電子署名を利用したメール認証技術との違い　メールで電子署名を利用して認証する技術には，DKIM 以外にも S/MIME[*3] や OpenPGP[*4] があります．いずれもテキスト以外の情報をメールに取り込む MIME の仕組みを利用して，MIME パートに電子署名を取り込んだり，メッセージパート部分を暗号化し，復号化のための情報を別の MIME パートに示す仕組みです．電子署名には，DKIM と同様に公開鍵暗号技術を利用するため，S/MIME によるメールを送信する前に予め公開鍵に関する情報を渡しておく必要があります．OpenPGP も基本的には同じ手順が必要です．ここでは，S/MIME を例に DKIM との違いについて解説します.

S/MIME の歴史は古く，最初の仕様 (Version 3) が作られてから 20 年以上が経過しています．にもかかわらず，実用的に普及が進んでいるとは言い難い状況であるのには，理由があります．S/MIME は，メール送信者とメール受信者との間，End-to-End で共通した仕様によって基本的には利用される技術です．そのため，まず MUA やウェブメールシステム等などメール利用者が直接利用するシステムで機能が実装され，さらにそのメールの送受信者の間で利用することや方式が一致していることが前提です．その上でメールを送信する際には，メール送信先毎に証明書を予めメール等で受け取っておくなど，事前に入手する必要があります．メール受信時にもメール送信者が S/MIME に対応しており，メール受信者の証明書を既に保持していることによって，電子署名や暗号化の機能が利用できることになります．こうしたやりとりは，メールを送受信する相手の数だけ必要であり，相手毎に予め証明書を入手しておく必要もあります．証明書の信頼性を高めるためには，信頼性の高い認証局が発行した証明書の利用が推奨されますが，例えばメール利用者の数だけ証明書を発行する必要があるとすれば，かなりのコストを伴うことになります．また，安全性の観点から一定期間毎に証明書を更新する必要がありますので，導入時の一時的なコストということにもなりません．また，暗号化に利用した場合，受信時の処理方法にもよりますが，復号化せず暗号化したままメッセージを保管していた場合，復号のための鍵を更新して古いものを保持しない運用をしていた場合，元のメッセージを参照することもできなくなってしまいます.

DKIM は，基本的にメールサーバ間で電子署名の作成と検証を行いますので，個々のメールの利用者が何か新たな設定や作業は必要ありません．電子署名の検証に必要な公開鍵は，DNS の仕組みを利用して取得で

[*3] 最新の仕様は RFC8550(Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling) と RFC8551 (Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification)

[*4] RFC4880 (OpenPGP Message Format)

きますので，たとえばメールサーバ間で予め受け渡し等をしておく必要もありません．

最近では，こうしたDNSを利用して公開鍵の受け渡しをするプロトコルが増えており，メールサーバ間での通信の暗号化を行うTLS(STARTTLS)では，DNS上のTLSA資源レコードに証明書に関連した情報を設定するDANE(DNS-based Authentication of Named Entities)があります．S/MIMEやOpenPGPについても，DNS上にSMIMEA，OPENPGPKEY資源レコードを利用する仕組みが提案されています．

2.2.4 DMARC の設定

DMARC レコードを設定する際に重要なことは，ポリシー (p) タグの内容を検討することです．ポリシーには，none，quarantine，reject の 3 種類の値のうちいずれかを設定できます．メール受信側でなりすましメールを受け取らないようにするためには，より強いポリシーの値（quarantine もしくは reject）を設定すべきです．しかしながら，受け取ってもらうべき正規のメールが，SPF と DKIM で認証できない場合があったり，認証したドメイン名がヘッダ上の RFC5322.From のドメインと組織ドメインも含めて異なる場合がある場合は，メールが届かなくなってしまう場合があります．より強いポリシーを DMARC レコードに設定する場合は，DMARC レポートを受信し認証結果を参照することで，受信側での認証状況を事前に確認しておくべきです．

■DMARC レポートについて　DMARC レポートの送信先は，DMARC レコードに集約レポート（aggregate report, rua タグで指定）や失敗レポート（failure report, ruf タグで指定）それぞれに対するメールアドレスとなります．これらのメールアドレスが記述される DMARC レコードは，DNS の仕組みにより誰でも参照できる情報です．また，ドメイン名がわかれば，そのドメイン名に対する DMARC レコードの設定される場所も容易に類推できます．そのため，レポートの宛先に対しては，DMARC レポート以外にも不要な迷惑メールや偽の情報を含んだ悪意ある DMARC レポートが送信される可能性があります．DMARC レコードに記述するレポートの宛先（メールアドレス）では，通常のメール受信と同様に，送信ドメイン認証技術で認証したり必要に応じて悪意ある送信元であるかどうかを判断するなど，正しいレポートだけを受信できるよう対策をしておくべきです．

2.2.5 メールに利用しないドメイン名に対する設定

ドメイン名を取得したとしても，それを Web サイトにだけ利用するなど，メールに利用しない場合もあります．しかし，ドメイン名が存在（例えば A 資源レコードが参照できる等）する場合，それが送信ドメイン名に悪用される場合があります．こうした実在するドメイン名を悪用し，なりすましメールを送らせないための対策としても，送信ドメイン認証技術を利用することができます．

■送信ドメイン認証技術の利用　まず，SPF ではメールを送信しないドメイン名，という意味を示す SPF レコードの設定ができます．

```
example.jp.  TXT  "v=spf1 -all"
```

この設定では，SPF 認証が pass する送信元の IP アドレスの指定が無いため，全て末尾の-all に適合します．prefix が-（ハイフン）なので，いかなる送信元からのメールであっても認証失敗となります．

続いて，DMARC レコードとして以下を記述します．

```
_dmarc.example.jp.  TXT  "v=DMARC1; p=reject"
```

例で示した example.jp ドメイン名は，SPF 認証が失敗しますので，そのドメイン名での DKIM 署名がなければ，DMARC としても認証失敗します．上記の設定では，DMARC レコードのポリシーが reject ですので，DMARC ポリシーに基づいた受信処理を実施している場合，その受信側に届かないことになります．

結果として，example.jp ドメイン名のなりすまし（詐称）利用を防ぐことができます．また，このようなメールに利用しないドメイン名をなりすましている送信者の情報については，DMARC レポートの設定（rua タグまたは ruf タグ）をすることで得ることができます．

■MX レコードへの設定　送信ドメイン認証技術とは直接関係ありませんが，ドメイン名のなりすまし利用を防ぐもう一つの方法があります．null MX と呼ばれる MX 資源レコードの設定方法で，RFC7505 で規格化もされています．

```
example.jp.  MX  0 .
```

上記の通り，MX 資源レコードの優先度を示す数値 (preference number) として，最も優先度の高い 0 を指定し，さらに受け取り先を示すラベル（ホスト名）を長さ 0 に設定します（設定方法によっては ‘‘.’’）．この設定により送信先のメールサーバは，宛先ドメイン名の配送先が見つからない (Null MX) と判断することができます．これにより，メール送信時に送信メールアドレスが実在するかを確認するような送信メールサーバや，メール受信時に送信ドメイン名の MX レコードを確認するような受信メールサーバで，null MX であるかどうか判断すれば，送信ドメイン名がなりすましているかどうかを判断することができます．

送信メールサーバは，メール配送に際し宛先ドメイン名に対して MX 資源レコードを参照します．MX 資源レコードが取得できなかった場合には，A あるいは AAAA 資源レコードを参照し，SMTP の接続先の IP アドレスを得ようとします．上記のように，null MX レコードを設定しておけば，その後の A あるいは AAAA 資源レコードを参照することなく，送信処理を中断することができ，送信メールサーバにとっても必要の無い DNS 参照を軽減することで，処理の負荷を下げることができます．

2.3 メール受信側の設定

メール受信側で送信ドメイン認証を行うためには，SPF, DKIM, DMARC それぞれの認証機能を受信側のメールサーバに組み込む必要があります．DMARC 認証を行うためには，SPF, DKIM それぞれの認証結果が必要ですので，事前に認証しておく必要があります．

いずれの送信ドメイン認証技術でも，認証されたドメイン名は，間違いなくその送信元であることを示しているだけですので，認証が通った（pass した）メールだから受け取るべきメールである，とは必ずしもいえません．迷惑メール送信者が，取得した独自のドメイン名に正しく送信ドメイン認証の設定を行い，認証するメールを送信していることも十分考えられますし，実際にそうしたメールも多く存在します．そうした一方で，認証が通ったメールはそのドメイン名は詐称されていないわけですから，その認証したドメイン名から受け取るべきかどうかを判断することはできます．但し，よくあるドメイン名と字形的に紛らわしい誤読してしまうような文字列を利用している場合も考えられます（`m` の代わりに `r` と `n` を使う `rn` など）ので，視覚的な判断に頼るだけでは十分とはいえません．こうした認証したドメイン名を評価し受け取るべきかどうかを判断することが，送信ドメイン認証技術の活用方法の一つです．

こうしたドメイン名に対する判断の基準（評価）を，ドメインレピュテーションといいます．紛らわしいドメイン名には，視覚的なもの以外にも何らかの時限的なキャンペーン（*ex.* example2020.jp）を連想するような数値等を既存ドメイン名と組み合わせる方法など，様々ななりすまし手法があります．こうしたドメイン名が取得できないような対策を，ドメイン名のレジストラ側も実施すべきですし，そうした基準があいまいなドメインレジストラは，それ自体の評価にも影響するようになるでしょう．認証したドメイン名に対するドメインレピュテーションについても，今後検討していくべきと考えます．

ここでは，それぞれの認証時に注意すべき事柄について，まとめます．

2.3.1 SPF 認証

SPF 認証には，メールの送信元（受信するメールサーバへの直接の通信元）の IP アドレスが必要です．そのため，一番最初に受信するメールサーバ上で SPF 認証する機能を組み込む必要があります．但し，一番最初に受信するメールサーバから何らかの方法で送信元 IP アドレスを受け渡しできる場合（例えばメールヘッダ上の `Received:` ヘッダが付与する情報で判断できる場合等）には，最初のメールサーバ以降でも認証できる場合があります．利用する SPF 認証機能について，確認しておく必要があります．

■SPF レコードの意味的判断　SPF レコードは，単なる IP アドレスやネットワークアドレスだけではなく，ホスト名やマクロを利用した記述などもできることから，送信元の範囲がわかりづらい場合があります．そのため，実際は送信元 IP アドレスの範囲が広大であったり，どんな送信元でも認証が通ってしまうような SPF レコードであるかもしれません．こうした SPF レコードが設定されているドメイン名は，Botnet 等を利用したなりすましメールの送信ドメイン名に利用されている可能性があります．

メール受信側としては，SPF の認証が成功 (pass) したか失敗したかといった結果だけでなく，可能であればこうした SPF レコードの内容に基づく意味的な観点からの認証判断ができると良いでしょう．

■DNS への過度な負荷の軽減　SPF 認証のために，DNS への問い合わせが通常は複数回行われます．そのため，DNS 側への負荷を過度に高めることを目的とした DoS 攻撃，あるいは広範囲なメール受信者を利用し

た DDoS 攻撃などに SPF 認証が悪用（そうした目的を持った SPF レコードの設定）されることも考えられます．

SPF では DNS の参照回数に上限（10 回）が設けられていますが，その範囲内でも特定のドメイン名に過度の負担を強いるような攻撃が行われる可能性があります（送信量や設定している SPF レコードの内容等）．メール受信側で，こうした DoS 攻撃元となることを予測して防御することは簡単ではないですが，何かあった場合に DNS 参照を抑制できるような仕組みを用意しておくことは有益です．

■DNS を利用した受信側への負担　上記は，SPF レコードを設定する側が，関係のないドメイン名等に対する攻撃の手法ですが，DNS を管理する側で，意図的に応答を遅くしたりタイムアウトさせるなどの手法によって，SPF 認証するメール受信側に対して，認証処理の負荷を増やす方法も考えられます．

大量にメールを受信するメールサーバにとって，受信処理するメールが多く滞留してしまうことは，通常はメールサーバの負荷を増やすことになります．メール受信側としては，こうした異常状態が発生した場合に検知できる仕組みがあると対策がたてやすくなります．

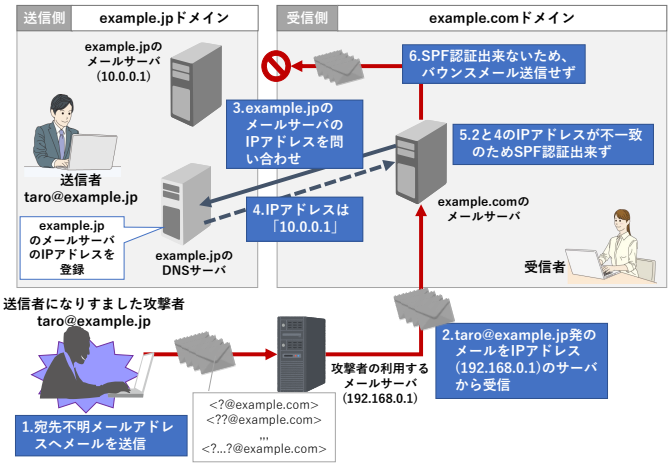

図 2.2　バックスキャッタと SPF 認証による抑制

■バックスキャッタ対応のための SPF 認証　メール配送のプロトコル SMTP では，一旦受け取ったメールが宛先不明等の理由により配送できなかった場合，送信元にエラーメール (Non-Delivery Notification) として返信（バウンス）します．現在では，こうしたバウンスメールの仕組みが，余計なトラフィックを増やしたり，これを悪用した迷惑メール送信や攻撃（バックスキャッタ）が発生しています．例えば，エラーメールを送信する受信側に対して，意図的に宛先不明となるメールアドレスを設定し，大量にメール送信することで，エラーメールの送信先に対して大量のエラーメールを送信し DoS 攻撃することができます．

バウンスメールの送信先は，SMTP 上の送信者，つまり RFC5321.From であり SPF が認証するドメインのメールアドレスです．メール受信側がバウンスメールの送信時に，元のメールの SPF 認証結果を利用し，認証できなかった（pass ではない）場合にバウンスメールを送信しないという仕組みができれば，バックスキャッタ攻撃を抑制できますし，不要なトラフィックを減らすことができます．

メール転送時に，転送先では SPF 認証が失敗してしまうため，転送先でバックスキャッタ対策としての SPF 認証結果利用を実施すると，本来エラーメールが送られる正規のメールに対しても，エラーメールが抑制されてしまう，という危惧も考えられます．しかしながら，そもそも転送先に配送できなかったメールにつ

いて元の送信者にエラーメールとして送信することの是非もあるので，こうしたエラーメールを抑制したとしても影響は限定的と言えます．さらにこうしたバックスキャッタの懸念等から，そもそもエラーメール自体を全く送信しない受信側も増えています．エラーメールは，ちょっとした宛先指定の間違いを気づかせることもできる機能ですので，全く送信しないよりは，現段階では限定的に活用していくべきと考えます．また，メール受信の最初のメールサーバで，宛先が存在するかどうかを確認することができれば，その処理（SMTP 上の）の途中でエラーコードを返すことができるはずですので，そもそもエラーメールを送信する必要もありません．こうした仕組みが実現できるかを検討するべきでしょう．

2.3.2 DKIM 認証

DKIM 認証は，メールの内容から作成される電子署名を検証することで認証を行います．そのため，メールの内容が電子署名作成時から変更されていなければ，いつでも DKIM での認証ができます．ただし，DKIM の認証（電子署名の検証）には公開鍵の情報が必要になるため，DNS で公開されている DKIM レコードにアクセスできる間は認証可能，ということになります．そのため，なるべくメール受信時に DKIM 認証するか，DKIM レコードを取得し保存しておく必要があります．

■署名作成方法の確認　DKIM 認証は，メールヘッダに付けられた DKIM 署名（電子署名）を検証することによって行われます．メール受信側では，この電子署名の作成方法についても注意する必要があります．

DKIM 署名の対象は，`DKIM-Signature` ヘッダの `h` タグで指定されたヘッダと，メール本文です．署名対象に含まれるメールヘッダとして，`From` ヘッダは必須となっています．メール受信者にとってもメールの送信者と考える情報であり，DMARC にとっても認証対象となる情報です．そのため DKIM 署名の検証の際は，`From` ヘッダが改変されていないことを検証するうえでも，`h` タグに `From` ヘッダが含まれていることを確認すべきです．

署名対象のメール本文についても注意が必要です．署名対象のメール本文の長さは `l` タグで指定します．指定する数値は Octet 値 (いわゆるバイト数) です．本文の長さ `l` タグを省略した場合は，メール本文全体になりますが，近年は添付ファイル等の利用も増えておりメールサイズも大きくなっているため，本文全体を署名対象に含めるような設定は非効率な場合があります．しかしながら，署名対象の長さの値が小さすぎる場合には，その長さ以降の本文を改変した場合でも検証できてしまいますので，ある程度の長さを設定しておくべきです．特に，次項で述べる replay attacks（再演攻撃）でも悪用される可能性がありますので，署名対象となる本文の長さにも注意すべきです．

■再演攻撃対策　再演攻撃 (replay attacks) とは，DKIM 署名を作成するメールサーバからの送信メールを再利用し，別のメール受信者に大量送信等することで，最初の署名ドメイン名のレピュテーション等を利用して届きやすくしようとするメール送信攻撃です．例えば，大手のフリーメールを利用して自分自身に送信したメールを再利用すれば，最初のフリーメールのメールサーバを経由させなくても，任意の宛先に大量に送信することができます．フリーメール側で迷惑メール送信防止策として送信数制限を設定していても，その制限によらず送信することができます．また，DKIM 署名付きの重要なメールを手に入れることができてその本文が短かった場合，その本文以降を利用して replay attacks により重要なメールを悪用することもできるかもしれません．

こうしたことからも，メール送信側は署名対象のメールヘッダと本文については何らかの配慮が必要であり，メール受信側は簡単に悪用されそうな状態の電子署名だったり，同じ電子署名のメールが大量に届いたり

する場合には，SPF の認証結果や送信元のドメイン名を確認するなど，別の方法を組み合わせるなどの対策が必要です．

2.3.3 DMARC 認証

DMARC は，SPF や DKIM で認証されたドメイン名とヘッダ上の送信者（`From` ヘッダ, RFC5322.From）が一致か同じ組織ドメインであるかどうかを認証します．組織ドメインとは，ドメイン名の登録事業者に登録されるドメイン名のことですが，実際には TLD(Top Level Domain) ごとにルールが異なっており，統一されたルールで組織ドメインを決めることは難しいため，現時点では public suffix リストを用いて判定することになっています．

DMARC では，ヘッダ上の送信者のドメイン名を認証しますが，なりすましメールや迷惑メールの送信者は，ヘッダ上の送信者欄（`From` ヘッダ）の補助情報であるディスプレイネームを悪用します．例えば，任意の文字列が設定できるディスプレイネーム部分になりすますメールアドレスやブランド名などの表示名を設定し，DMARC の認証対象であるディスプレイネームに続くメールアドレス (`<local-part@domain>`) 部分に，DMARC 認証が pass する全く別のドメイン名を設定することで，なりすまそうとしているドメインがあたかも認証できたように思わせようとします．これは，メールソフト (MUA) や Web Mail などが，利用者の利便性のためにディスプレイネーム部分を優先して表示する場合が多いことも悪用の要因となっています．

```
Subject: 請求書の送付
From: forged_mail_address@example.jp <real@example.com>
```

■認証ドメインの評価判定　DMARC に限らず SPF や DKIM でも同様ですが，送信ドメイン認証の結果が pass であっても，そのメールが迷惑メールではないということではもちろんありません．ただこれらの送信ドメイン認証技術で認証されたドメイン名については，送信者として識別できる情報であるということは，現在のところ十分に信用できる情報であるといえます．これら認証されたドメイン名，つまり送信者の情報を用いて，受け取るべきメールであるかどうかを判断することはある程度可能なはずです．この認証されたドメイン名を利用した判断，つまりドメイン名の評価の基準をレピュテーションと呼びます．

現在のところ，ドメイン名のレピュテーションに関する明確な基準といったものや，広く使われているサービス等はまだ無いと思いますが，今後の送信ドメイン認証技術の普及とともに迷惑メール判定の手法として期待される分野です．

■DMARC ポリシーと受信処理　DMARC レコードでは，DMARC ポリシーとして 3 つの値 (`none`，`quarantine`，`reject`) を設定することができます．これらの値は，ドメイン管理者がメール受信者に求める動作ですが，一般的に `none` 以外の 2 つがより強いポリシーと考えられています．また設定できるポリシーの中では，`none` や `reject` は理解しやすい動作ですが，`quarantine` は隔離の意味ですが一般的にそうした機能を持たないメール受信側は多いと思います．`quarantine` の意味するところは，DMARC 認証が失敗 (`fail`) した場合，そのメールは怪しいものとして取り扱うことを受信側に求める，メール受信者にそう思わせることを求める設定です．つまり例えば，迷惑メールフォルダといった機能があるのであればそこに保存するといった動作や，`[spam]` といった表示や何らかのマークをそのメールに付けるような動作が期待されるものです．

■DMARC レポートの送信　ポリシーの表明以外の DMARC の特徴の一つに DMARC レポートがあります．これまでの SPF や DKIM などの送信ドメイン認証技術では，送信側での設定が正しく機能しているのかを確認する方法がありませんでした．そのため，例えば SPF レコードの記述内容にフォーマットとしての間違いが無いか，設定したメールサーバ以外のところから正規のメールが送信されていないか，DKIM の電子署名が正しく作成されているか等について，送信側からでは把握することができず，また効果も見えづらいことなどによって，送信ドメイン認証技術の設定を躊躇する要因の一つと考えられてきました．

DMARC レポートでは，メール受信側からドメイン管理者側（正確には DMARC レコードに記述されているレポート送信先）へ，集約 (aggregate) レポートであればメール送信元（IP アドレス）や SPF, DKIM, DMARC の認証結果などを伝えます．これらの DMARC レポート情報から，正規に送信されたメールの認証状況や，それ以外のドメイン名をなりすますメールの送信数や送信元の情報を得ることができます．こうした DMARC レポートを多くのメール受信元から受け取ることで，送信ドメイン認証技術の設定状況や，なりすましメールの状況などをドメイン管理者は把握できるようになります．

これらの DMARC レポートは，例えば集約 (aggregate) レポートであれば，メール受信側が受信メールの状況を記録し，定期的に集計して DMARC レコードにレポート送信先を設定しているドメイン管理元に送信する必要があります．メール受信側にとっては，本来行うべきメール受信以外に必要となる新しい処理であり，受信側にとっては直接的には有益ではない処理と捉えられるかもしれません．しかしメール送信側やドメイン管理者側にとって DMARC レポートは，送信したメールがどのように認証されているか，管理しているドメイン名がどの程度なりすまされているかを知る有益な情報です．メール受信側が，DMARC レポートを送信することでメール送信側がこれらの情報を得ることができ，このことが送信側で送信ドメイン認証技術を導入する動機付けにもなり，受信側にとって送信ドメイン認証技術に対応したメールが増えることで，なりすますメールが検知しやすくなったり，ドメインレピュテーションを用いることでメールの振り分けができるようになるわけです．こうした循環となるよう，メール受信側での DMARC レポートを送信すべきです．

DMARC レポートの送信については，悪意ある DMARC レコードに注意する必要があります．関係の無い第三者に DMARC レポートが大量に送信されることで，DoS 攻撃とならないよう，DMARC レポート受信の委譲が設定されているか，送信頻度は適切か，失敗 (failure) レポートを送信している場合，それらが悪用されていないか等，ある程度の確認も必要となります．

2.4　メール再配送の課題

最初のメール作成者から，最後に届けられるメール受信者へ直接メール配送が行われない場合，最後のメール受信者での送信ドメイン認証技術 SPF, DKIM, DMARC が正しく認証できない場合があります．このようなメールの配送先が異なるようなメール配送の流れを，ここではメール再配送 (indirect email flow) と呼びます．こうした課題は，配送の途中に存在する中間者の処理の仕方によって，認証結果も異なってきます．ここでは，それぞれの再配送の場合ごとに，認証が正しく行われない原因とその対策について述べます．

2.4.1　メール転送

最初のメール送信者が，宛先に指定したメールアドレスに配送された後に，別のメールアドレスに配送するメール転送の仕組みが，メールがインターネット上で利用された初期の頃から利用されてきました．具体的には，メールボックス（メールの宛先）で".forward"（ドットフォワード）ファイルに転送先メールアドレスを記述するメール転送や，メールの受信側で実際の宛先を設定するエイリアス機能などです．

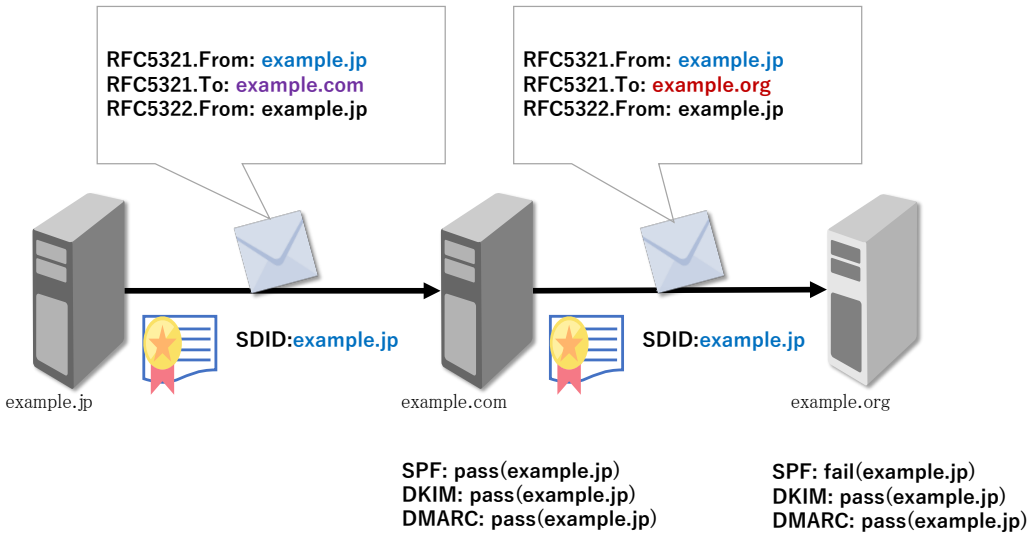

図 2.3　メール転送

メール転送による再配送の場合，メール本文の変更はほぼ行われず，メールヘッダ上の `Received:` ヘッダに経由したことが記される程度で転送先に送られます．この場合の送信者情報は，SPF の認証対象の RFC5321.From (envelope from) は，最初のメール送信者のままとなり，宛先の RFC5321.To のみが転送先と指定されたメールアドレスとなって転送先に配送されます．最終的な転送先 (example.org) では，SPF の認証対象となるドメインが RFC5321.From のままであるため，最初のメール送信者 (example.jp) を認証しようとしますが，転送先 (example.org) の配送元は，転送元 (example.com) であるため通常は SPF の認証が失

敗します.

最初のメール送信者 (example.jp) が DKIM 署名を付けていれば，転送でメール本文等はほぼ改変されませんので，最終的な転送先 (example.org) でも DKIM 署名は検証できます．この場合，DKIM の署名者 (SDID, Signing Domain Identifier) とメールヘッダの送信者 (RFC5322.From) のドメインが一致するか同じ組織ドメインである場合は，DMARC の認証も pass することになります．つまり，メール転送が行われる可能性があるメール配送経路については，最初のメール送信時に DKIM 対応（署名付加）しておけば，転送先でも正しく DMARC 認証できることになります.

2.4.2 メーリングリスト

メーリングリストは，特定のメールアドレスに届いたメールを，登録された複数のメールアドレスに再配送する仕組みです．前述のエイリアス機能を使った簡易的な方法でも実現できますが，現在では専用のソフトウェア[*5]を利用した運用が一般的となっています.

メーリングリストソフトウェアは，再配送先のメンバの管理（宛先不明となったメンバを配送保留や削除等を行う）を行うため，投稿者 (Submitter) から送信されたメールを一旦受け取り，エラーメールを受け取るために RFC5321.From (envelope from) をメーリングリストを運用しているドメインに設定し，メーリングリストのメンバ (Subscriber) 宛てにそれぞれ配送します．メーリングリストとして SPF を認証させるためには，メーリングリストのドメイン名 (example.com) に SPF レコードを設定し認証できるようにします．最終的なメール配送先であるメーリングリストのメンバ (example.org) では，SPF の認証は pass しますが認証されるドメイン名はメーリングリスト (example.com) となります．そのため，メールのヘッダ上のドメイン名は最初のメール投稿者 (example.jp) のままのため，DMARC としてはドメイン名が一致しない (example.jp ≠ example.com) ため，DMARC 認証としては失敗します．投稿者側で DKIM 署名をしていれば，メール本文等の改変が無ければメーリングリストメンバ側では DKIM の認証ができますし，メールヘッダ上のドメイン名も同じですので，DMARC も認証できます.

しかしながら，現在のメーリングリストの使われ方の多くは，メールの件名（Subject ヘッダ）にメーリングリストであることを示す文字列を挿入したり，メール本文の最後にフッタとして文字列を追加するといった運用がよく行われています．メーリングリスト側で，こうしたメール改変の処理を行うと，投稿者の DKIM 署名が合わなくなり DKIM の認証が失敗します．これを回避するために，メーリングリスト側でメールの改変後に DKIM で再署名を行えば，DKIM としては認証できますが，署名を作成できるのはメーリングリストのドメイン名 (example.com) としてです．よって DMARC としてはヘッダ上の送信者のドメイン名と署名ドメイン名とが一致しなくなるため (example.jp ≠ example.com) 認証が失敗します.

こうしたメーリングリストによるメール再配送の課題を解決する方法としては，以下の三つの方法があります.

- メール内容の改変をしない
- メーリングリストドメイン名での DKIM 再署名と RFC5322.From の変更
- ARC の利用

DKIM 署名対象となっているメールヘッダや本文を変更しなければ，メーリングリストへの投稿者が付け

[*5] Mailman などが広く使われています

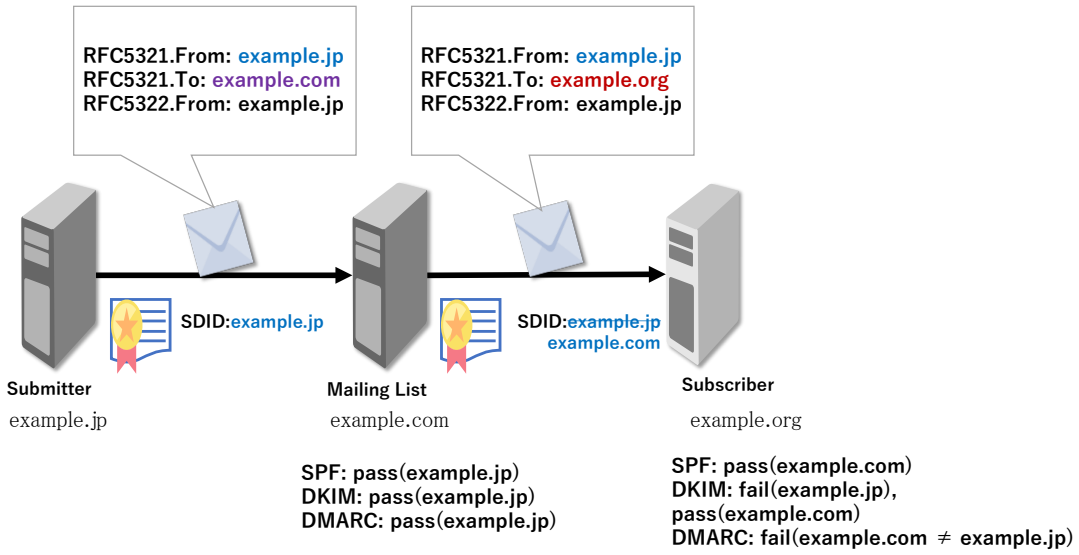

図 2.4　メーリングリストによるメール再配送

た DKIM 署名はそのまま検証できますので，メーリングリストメンバ側でも DMARC は認証できます．但し，これはメール転送によるメール再配送の課題でも同様ですが，最初のメール送信側で DKIM 署名に対応している必要があり，DKIM の普及に依存した解決方法と言えます．

メーリングリスト側でメールヘッダ上の送信者 (RFC5322.From) 情報をメーリングリストのドメイン名に変更し，DKIM で再署名すれば，メーリングリストのメンバ側では SPF も DKIM もメーリングリストのドメイン名 (example.com) で pass できますし，そのドメイン名は RFC5322.From のドメイン名とも一致しますので DMARC も pass します．この方法の利点は，メーリングリスト側の処理の変更だけで対応できる点で，メーリングリストへの投稿者やメンバ側は通常のメール送受信と変わらない動作となる点です．つまり，メーリングリストへの投稿者が必ず DKIM に対応していなくても良い（メーリングリスト側は別途正しい投稿者からのメールであるかを確認すべきですが）ですし，メーリングリストのメンバ側で送信ドメイン認証を導入していれば正しく送信ドメイン名を認証できることになります．一方で従来の運用方法と異なる点としては，メーリングリストメンバからみてメールの投稿者がわかりづらくなる部分です．こうしたデメリットを緩和するために，メーリングリスト機能側で最初の投稿者の情報を RFC5322.From のディスプレイネーム部分に残しておく処理も行われます．ただこの方法では，メール投稿者の最初のディスプレイネームの情報が失われてしまいますので，いずれにしても全く従来の方法と遜色が無いとは言い切れません．メーリングリストソフトウェアとして広く使われている Mailman では，こうした RFC5322.From の書き換えによる DMARC への対応方法の機能が最新版では利用できるようになっています．

ARC[*6]は，メールが再配送などメールサーバを経由する都度，認証を行いそれらを繋いでいくことで認証の連鎖 (Chain) を構成していく仕組みです．原理的には，メールを受け取る時にその送信元が信頼できる送信

[*6] The Authenticated Received Chain (ARC) Protocol, RFC8617

元である場合，その信頼する送信元が認証したその前の送信者も信頼できる，といった連鎖をつないでいきます．信頼の連鎖を構築するために，以下の 3 種類のヘッダが新たに提案されました．

AAR (ARC-Authentication-Results)： 従来の Authentication-Results ヘッダと同様の認証結果を示すものだが，ARC ヘッダセットを区別する Instance Tag（数値で示される）が追加された
AMS (ARC-Message-Signature)： 従来の DKIM-Signature ヘッダと同様に再署名に関する情報を示す
AS (ARC-Seal)： ARC 関連ヘッダを順番ごとに連結したデータから生成される電子署名

ARC は技術的には DKIM と同じ電子署名を利用しますが，仕組みとしては DKIM やそれを利用する DMARC とは関連しない別の仕組みとなっています．例えば，DMARC としては認証が失敗 (`fail`) する場合でも，ARC としては `pass` することがある，むしろそうした認証結果が不一致となる場合に信用の連鎖を用いてメールを認証するための新たな仕組みです．この信頼の連鎖を実現するためには，少なくとも直前のメール送信者（認証および署名者）が何らかの方法で信用できていることが前提となります．

送信ドメイン認証技術は，メール配送に関わるメールサーバ（送信側と受信側）それぞれでの導入によって認証が実現します．メーリングリストの課題に対する対策として 3 つの方法を述べましたが，これまでのメールの利用形態をなるべく維持したまま，メールシステム全体として変更する部分をなるべく少なくするためには，現時点ではメーリングリストサーバ側の修正（DKIM 再署名と RFC5322.From の変更）が望ましい手法といえるでしょう．

付録 A

関連 RFC

インターネット上で利用される様々な技術仕様は，IETF(Internet Engineering Task Force) が発行する RFC(Request for Comments) と呼ばれる番号によって区別された文章で示されます．RFC はその名称からもわかるとおり，技術は常に変化していく可能性があり，同じ技術仕様であっても様々な要因により変化していきます．RFC では，基本的に 1 度発行された RFC の番号はそのまま残され，仕様が改訂された場合には新たな番号が割り当てられます．全面的に仕様が改訂された場合には，新たな番号の RFC によって “obsolete” され，一部の仕様が更新された場合には “update” されます．

以下に，本導入マニュアルで重要な RFC について示します．

RFC No.	Title	概要	Updates
5321	Simple Mail Transfer Protocol	電子メールの配送プロトコル SMTP	RFC7504
5322	Internet Message Format	インターネットメッセージの形式	RFC6854
5965	An Extensible Format for Email Feedback Reports	電子メールのフィードバックレポートのための拡張形式	RFC6650
6376	DomainKeys Identified Mail (DKIM) Signatures	送信ドメイン認証技術 DKIM	RFC8301, RFC8463, RFC8553, RFC8616
7208	Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1	送信ドメイン認証技術 SPF	RFC7372, RFC8553, RFC8616
7489	Domain-based Message Authentication, Reporting, and Conformance (DMARC)	送信ドメイン認証技術 DMARC	RFC8553, RFC8616
7505	A ”Null MX” No Service Resource Record for Domains That Accept No Mail	メールを受け付けないドメインのサービス資源レコード Null MX	
8601	Message Header Field for Indicating Message Authentication Status	メッセージ認証状態を示すメッセージヘッダ	
8617	The Authenticated Received Chain (ARC) Protocol	認証された受信の連鎖 (ARC)	

RFC には，その仕様の性質により種別 (Category) があり，上記の RFC の Category も様々ですが，送信ドメイン認証技術を導入する上においては，いずれも重要な技術仕様です．

索引

送信ドメイン認証技術導入マニュアル第3版

2021 年 9 月発行

2023 年 2 月第3．1版発行

編集・発行　迷惑メール対策推進協議会

（事務局）

一般財団法人　日本データ通信協会　迷惑メール相談センター

〒170-8585　東京都豊島区巣鴨 2-11-1　ホウライ巣鴨ビル 7 階

TEL：　03－5907－5371

MAIL：q-meiwaku-mail-kyogikai@dekyo.or.jp

URL：　https://www.dekyo.or.jp/soudan/aspc/

印刷　一般財団法人　日本データ通信協会

迷惑メール対策推進協議会