＜5分でわかった気になる解説シリーズ＞
5分でわかるDMARC
5分でわかるARC
5分でわかるBIMI

北川 直哉

国立大学法人 東京農工大学
大学院工学研究院 先端情報科学部門

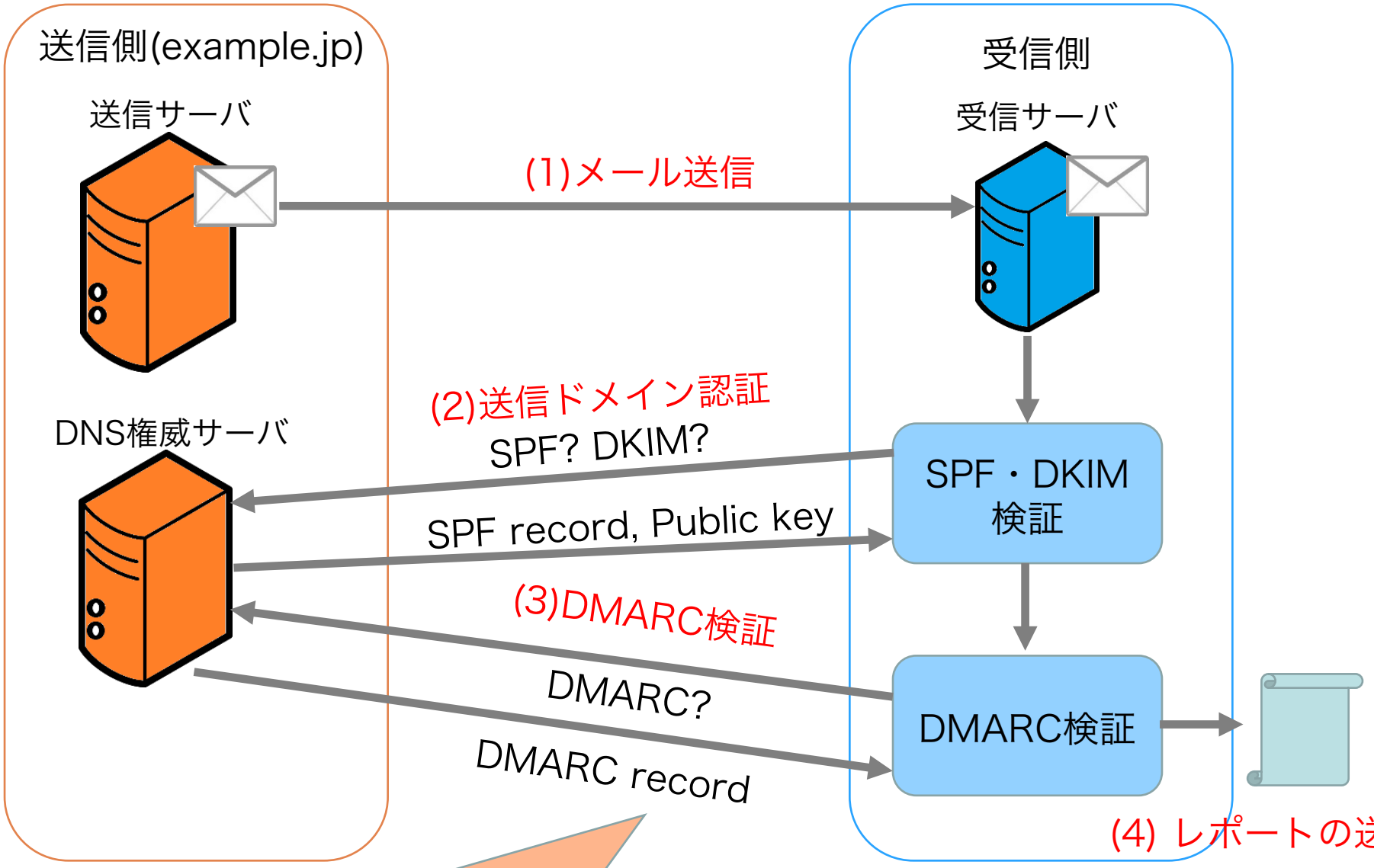
DMARC とは

- DMARC：
  Domain-based Message Authentication, Reporting, and Conformance

- ＜DMARCの2つの機能＞
  ① SPF/DKIM検証のエラー状況・統計情報のレポーティング機能
    - 自ドメインにおけるSPF/DKIM検証の効果
    - 自ドメインをなりすましたメールの状況

  ② SPF/DKIM検証失敗メールの取扱いを**送信側が**指定する機能
    - none（処理方法を指定しない）
    - quarantine（隔離する）
    - reject（受信拒否する）

受信側はポリシに基づいて
取扱い方法を決定


送信側(example.jp)
送信サーバ
受信側
受信サーバ
(1)メール送信
DNS権威サーバ
(2)送信ドメイン認証
SPF? DKIM?
SPF record, Public key
SPF・DKIM
検証
(3)DMARC検証
DMARC?
DMARC record
DMARC検証
(4) レポートの送信
v=DMARC1;p=reject;rua=mailto:ruareport@example.jp;
rua=mailto:ruareport@example.jp

DMARCのアラインメント

- SPF/DKIMでは,
  送信元ドメインと検証用ドメインは無関係でも良い
  （第三者署名を許可）
  - 東京農工大学の例：
    ・ヘッダFromドメイン＝cc.tuat.ac.jp
    ・DKIM検証ドメイン＝cctuat.onmicrosoft.com

正しいドメインの組を知る術がないため
偽の署名がついたなりすましメールも
検証に成功してしまう恐れがある

- DMARCでは第三者署名は認証失敗(fail)となる
  →より強固な送信ドメイン認証といえる

ARC とは

- ARC: Authenticated Received Chain
- IETF等で議論中の新しい仕組み (draft)
- DMARCはヘッダ上の送信元情報(From:)を認証するため,再配送メール(MLや自動転送)を正しく認証できない
  - 経由したサーバ度にReceived:ヘッダが追加される
  - つけられたReceived:ヘッダが正当かどうか検証不可

メールが再配送された場合でも
Authentication-Results:ヘッダを辿ることで
認証の連鎖を確認可能にする仕組み

ARC の仕組み

- 順番を示す番号(i=)を含むメールヘッダが追加

ARC ヘッダ	役割
ARC-Seal	ARC関連ヘッダを順番毎に連結したデータから生成される電子署名
ARC-Message-Signature	DKIMと同様の再署名情報
ARC-Authentication-Results	Authentication-Results:ヘッダの保存に使用

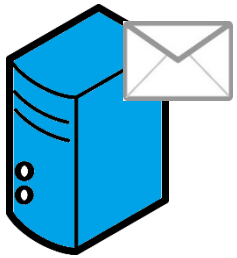
- 再配送時にARC独自に（DKIMとは別途）再署名
  →再配送先で認証の連鎖を検証し，ARC検証結果を取得
  （検証結果はARC-Sealヘッダに記録される）

再配送時におけるARC認証の例

sender@example.jp

ml@example.net

receiver@example.com

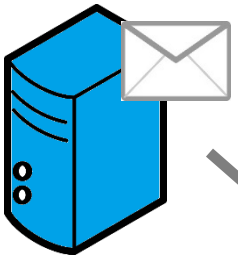


ML宛に
メール送信

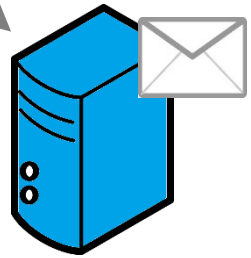
DKIM-Signature: v=1;…
From: sender@example.jp
To: ml@example.net
Subject: イベントのご案内

再配送時におけるARC認証の例

sender@example.jp　　ml@example.net　　receiver@example.com


ARC-Seal: i=1;…
ARC-Message-Signature: i=1;…
ARC-Authentication-Results: i=1;…
DKIM-Signature:v=1;…
From: sender@example.jp
To: ml@example.net
Subject: [ml:0123]イベントのご案内

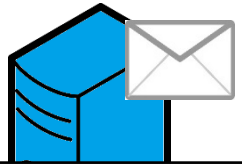



通常はDKIM=fail（subject変更）

再配送時におけるARC認証の例

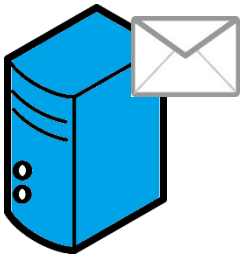
sender@example.jp

ml@example.net



receiver@example.com

Authentication results:…; arc=pass
ARC-Seal: i=1;…
ARC-Message-Signature: i=1;…
ARC-Authentication-Results: i=1;…
DKIM-Signature:v=1;…
From: sender@example.jp
To: ml@example.net
Subject: [ml:0123]イベントのご案内

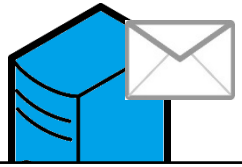


MLサーバが受信時にDKIM=passし,
ARCとして再署名することで
ARC=passとなる

再配送時におけるARC認証の例

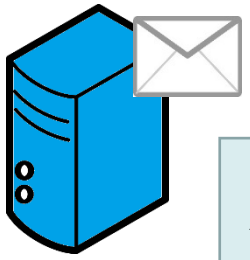
sender@example.jp

ml@example.net



receiver@example.com

Authentication results:…; arc=pass
ARC-Seal: i=1;…
ARC-Message-Signature: i=1;…
ARC-Authentication-Results: i=1;…
DKIM-Signature:v=1;…
From: sender@example.jp
To: ml@example.net
Subject: [ml:0123]イベントのご案内



受信者がさらに別のアドレスに自動転送していた場合，i=2の3つのヘッダが追加される

BIMI とは

- BIMI: Brand Indicators for Message Identification
- IETF等で議論中の新しい仕組み (draft)
- 送信ドメイン認証によって認証されたドメインからのメールをわかりやすく受信者に提示するための仕組み
- 認証されたメールはMUA(メールソフト)やウェブメールで，その送信者のドメインに関連した口ゴを表示（例えば企業の口ゴマーク）
- MUAが参照するための情報をメールヘッダに保存

BIMI の設定

- 送信ドメイン認証技術と同様にDNSを用いて情報を記述
  →メール送信側は受信者に表示させる自ドメイン配下の
  ロゴ等の情報をTXTレコードで示す

- 例：ヘッダFromがexample.comのBIMIレコードの設定

```
default._bimi.example.com IN TXT “v=BIMI1; f=png;
z=32x32; l=https://example.com/bimi/”
```

  - default：セレクタ（切替により同一ドメインで複数定義可能）
    - セレクタの切替は送信するメールヘッダで指定
    - 現在（仕様議論中）はBIMI-Selectorヘッダを使用
      （指定しない場合のデフォルト値はdefault）
    - 例：サブドメイン logo を使用する場合
      → BIMI-Selector: v=BIMI1; s=logo

BIMI の設定

- 送信ドメイン認証技術と同様にDNSを用いて情報を記述
  →メール送信側は受信者に表示させる自ドメイン配下の
  ロゴ等の情報をTXTレコードで示す

- 例：ヘッダFromがexample.comのBIMIレコードの設定

default._bimi.example.com IN TXT “v=BIMI1; f=png;
z=32x32; l=https://example.com/bimi/”

パラメータ	意味
v	バージョン番号(BIMI1)
f	イメージファイルのフォーマット(png, tiff, jpg, svg)
z	イメージのサイズ
l	イメージの場所（起点URIを指定）

BIMIの受信側での処理

- 受信側では，DMARCで認証できたドメインに対してBIMIレコードを取得
  - BIMI-Selector: ヘッダでセレクタが指定されている場合は指定されたドメインを利用してBIMIレコードを取得
- Authentication-Resultsヘッダに BIMIの結果情報を記録
  - ドメイン名（header.d=）
  - セレクタ名（selector=）
- BIMIレコードで指定されたイメージファイルの場所を BIMI-Location: ヘッダに保存

```
BIMI-Location: v=BIMI1; l=https://example.com/bimi/32x32.png
Authentication-Results: bimi=pass header.d=example.com selector=logo
From: sender@example.com
BIMI-Selector: v=BIMI1; s=logo
```