black hat
USA 2020
AUGUST 5-6, 2020
BRIEFINGS

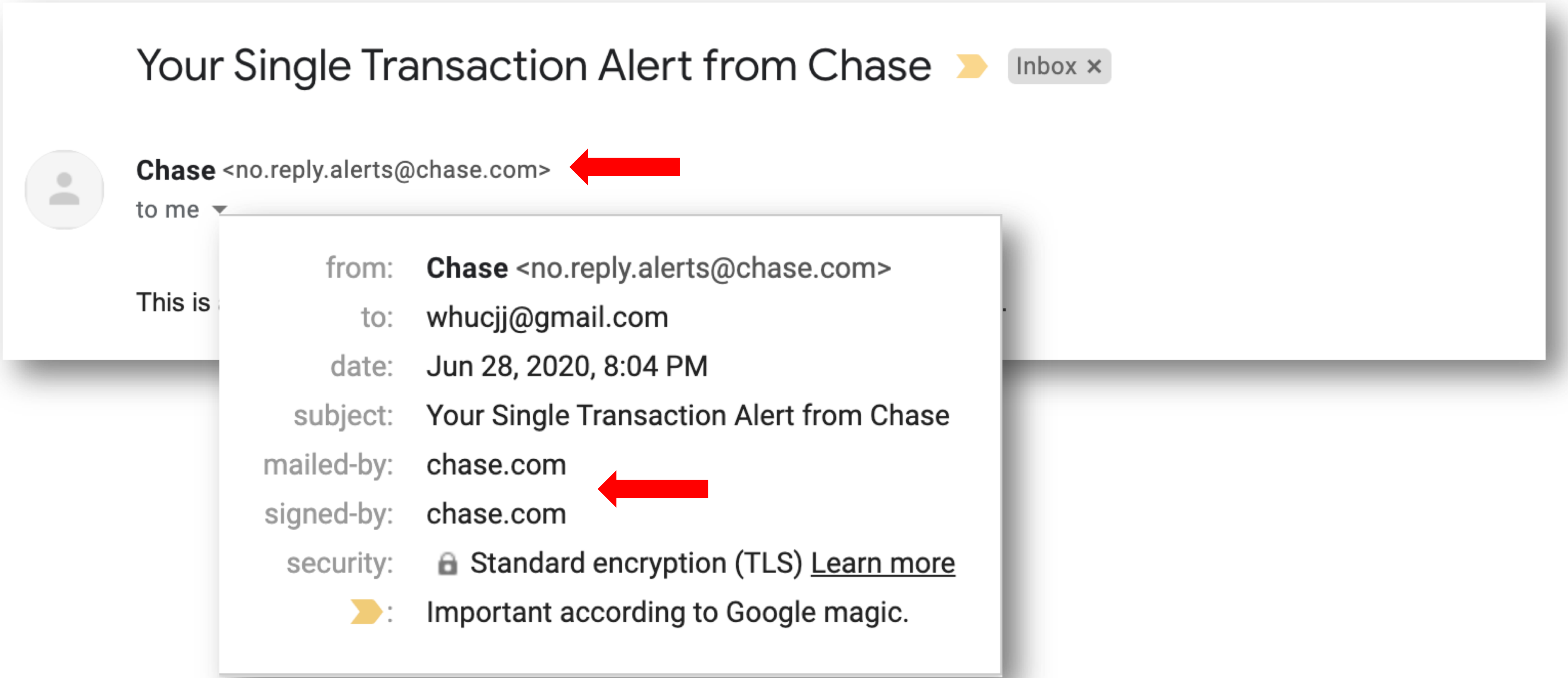
You Have No Idea Who Sent That Email: 18 Attacks on Email Sender Authentication

Jianjun Chen, Vern Paxson, Jian Jiang

About Us

- Jianjun Chen: Postdoc at ICSI
  - HTTP, Email: “CDN forwarding loop”[NDSS16], “Host-of-troubles”[CCS16]
- Vern Paxson: Professor at UC Berkeley
  - Creator of the Bro IDS
  - Co-founder of Corelight, providing network traffic analysis solutions
- Jian Jiang: Senior Director of Engineering at F5 (Shape Security)
  - DNS, Web: “Ghost DNS”[NDSS12], “Cookies lack Integrity”[USENIX15]

How Do You Verify the Email Sender?


Your Single Transaction Alert from Chase

Inbox ×

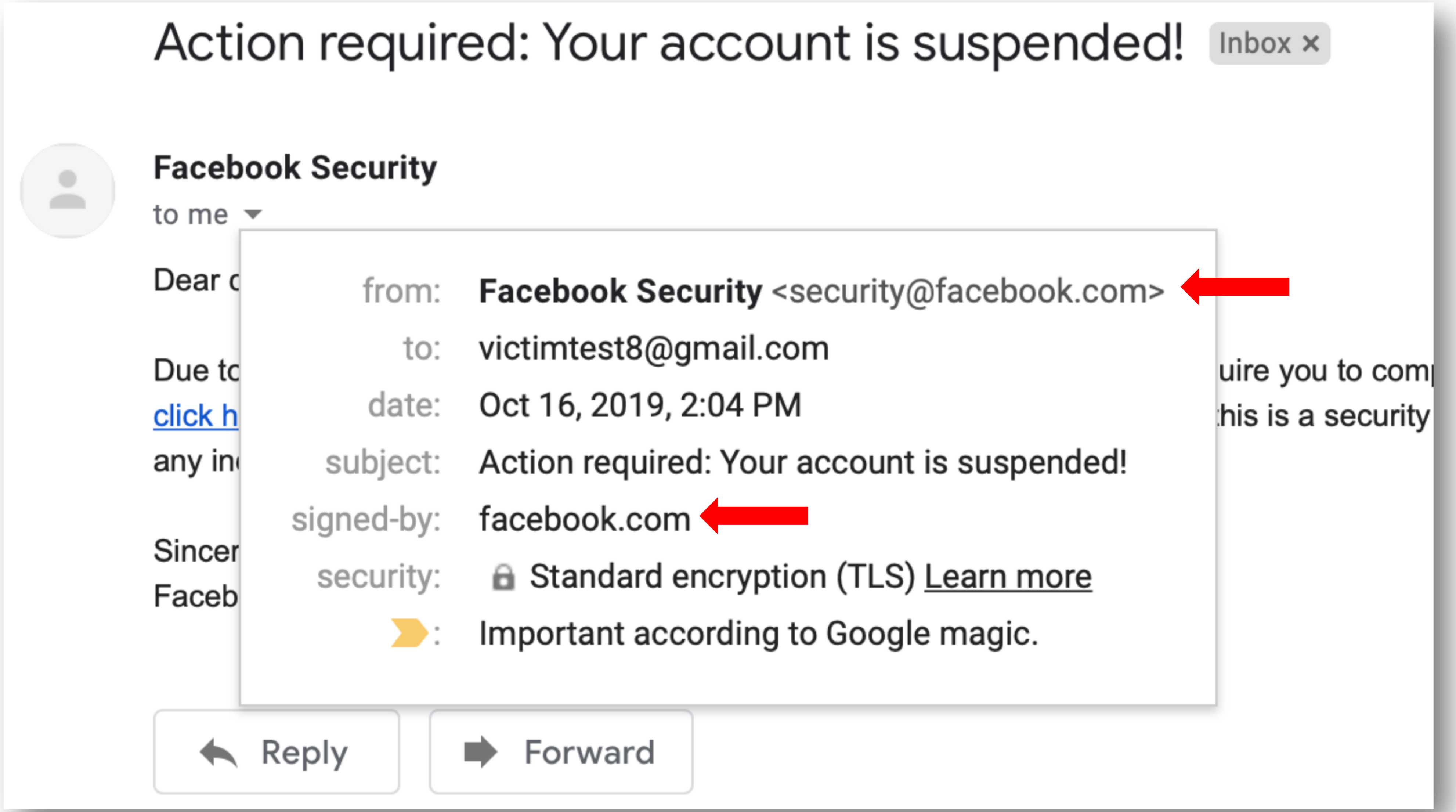
Chase <no.reply.alerts@chase.com>

to me

This is

from:	**Chase** <no.reply.alerts@chase.com>
to:	whucjj@gmail.com
date:	Jun 28, 2020, 8:04 PM
subject:	Your Single Transaction Alert from Chase
mailed-by:	chase.com
signed-by:	chase.com
security:	Standard encryption (TLS) Learn more
:	Important according to Google magic.

A Case of Our Spoofing Attacks on Gmail (Fixed)



Action required: Your account is suspended! Inbox ×

Facebook Security
to me

Dear c

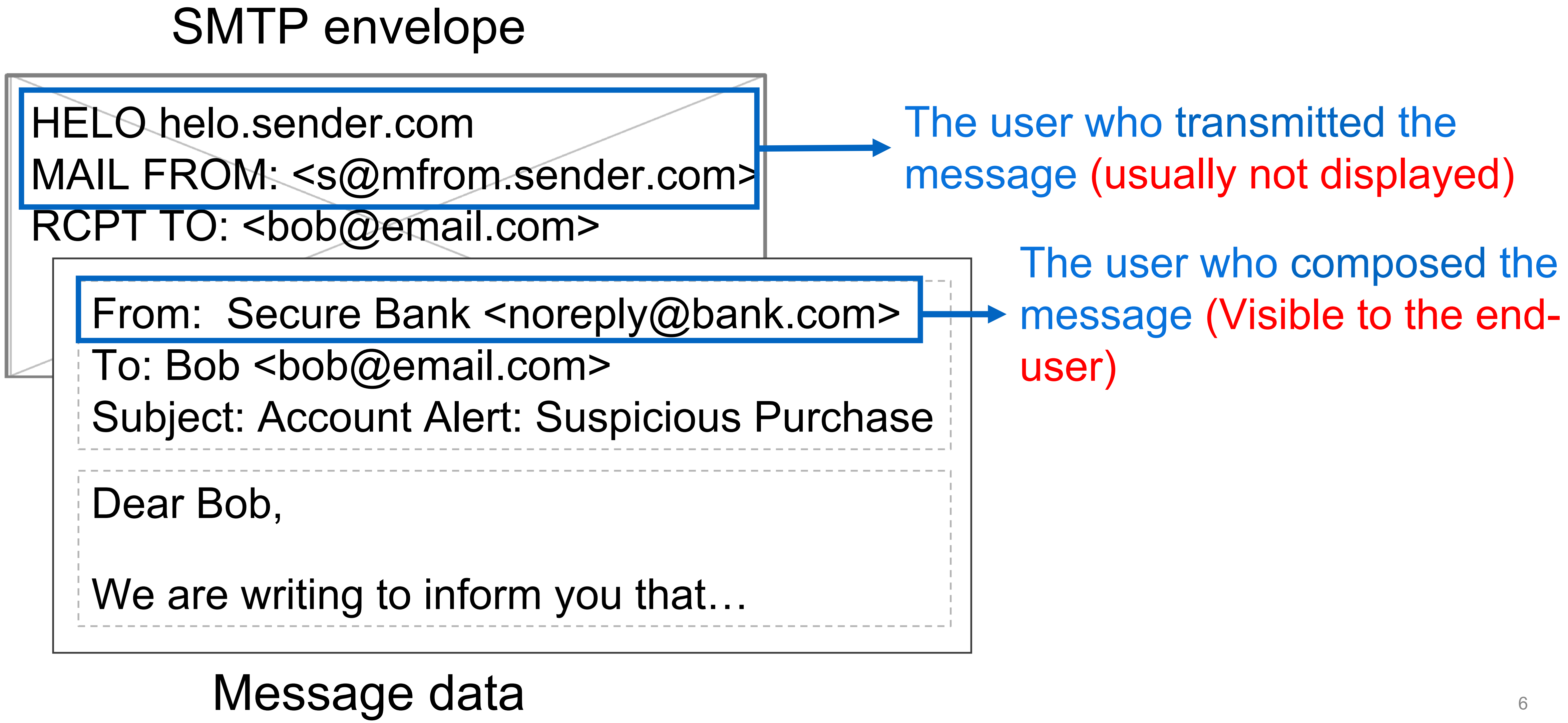
Due to … uire you to com
click h … this is a security
any in

Sincer
Faceb

from: **Facebook Security** <security@facebook.com>
to: victimtest8@gmail.com
date: Oct 16, 2019, 2:04 PM
subject: Action required: Your account is suspended!
signed-by: facebook.com
security: Standard encryption (TLS) Learn more
: Important according to Google magic.

Background:
Sender & Authentication

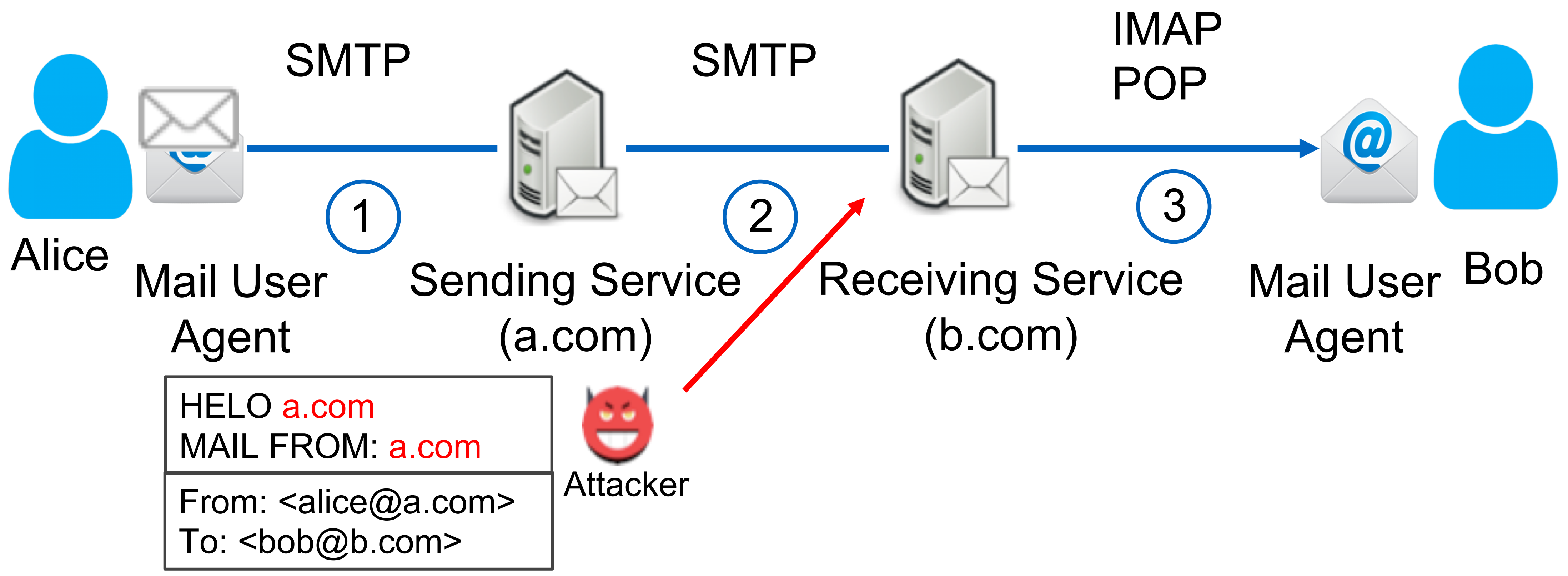
Background: Who's the Sender?


SMTP envelope
The user who transmitted the message (usually not displayed)
The user who composed the message (Visible to the end-user)
Message data

```
HELO helo.sender.com
MAIL FROM: <s@mfrom.sender.com>
RCPT TO: <bob@email.com>
```

```
From:  Secure Bank <noreply@bank.com>
To: Bob <bob@email.com>
Subject: Account Alert: Suspicious Purchase

Dear Bob,

We are writing to inform you that…
```

Background: Email Transmission



HELO a.com
MAIL FROM: a.com

From: <alice@a.com>
To: <bob@b.com>

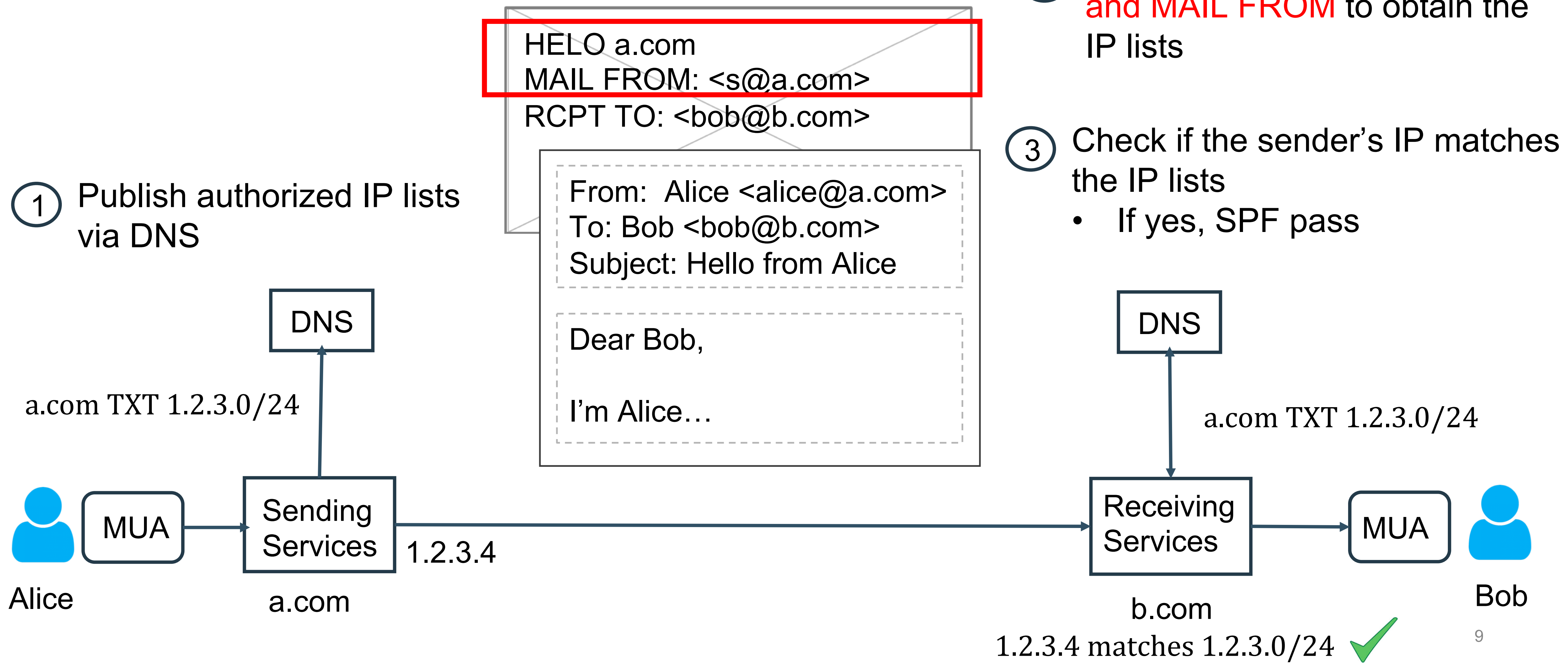
The original SMTP has no built-in authentication mechanism

- Anyone can spoof any identity in HELO/MAIL FROM and From

Three Sender-Authentication Protocols

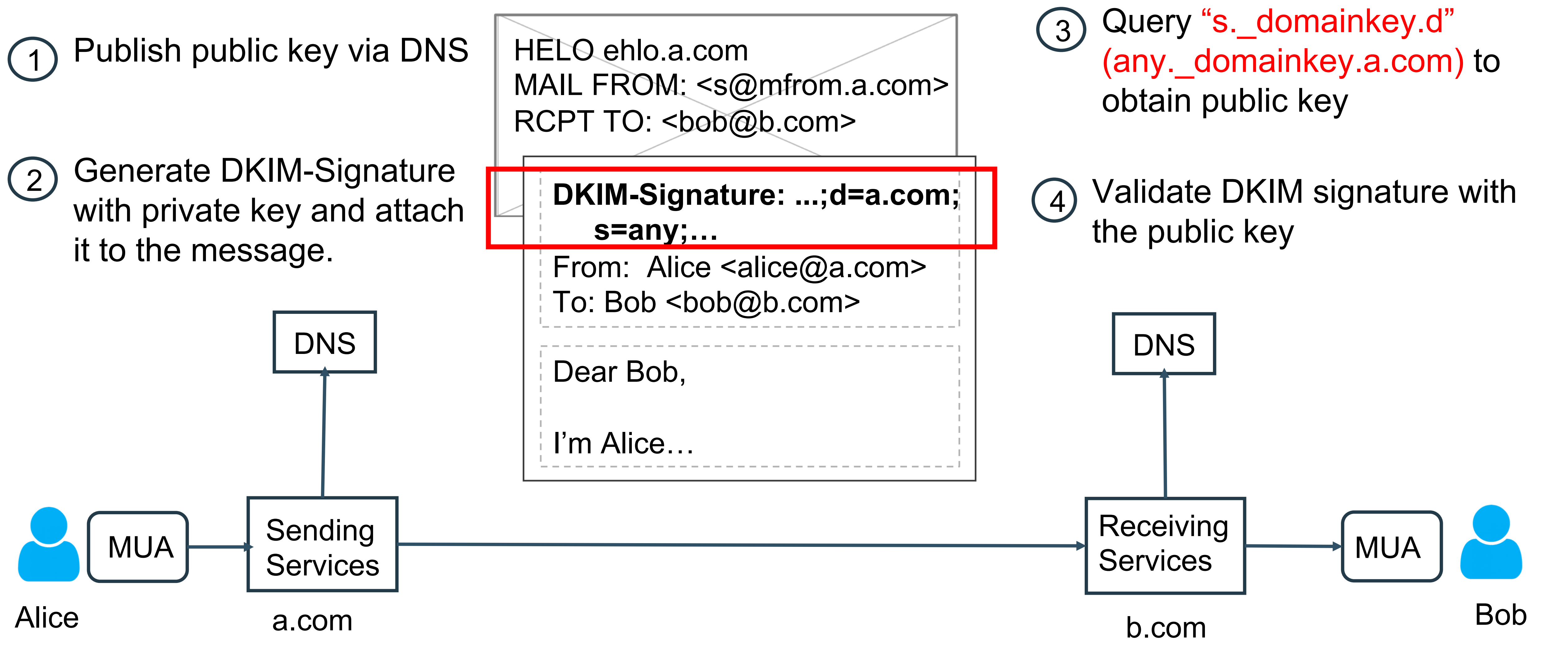
- Sender Policy Framework (SPF, RFC 7208)
  - verifying the IP address of the sending domain
- DomainKeys Identified Mail (DKIM, RFC 6376)
  - verifying the email is signed by the sending domain
- Domain Message Authentication, Reporting and Conformance (DMARC, RFC 7489)
  - “how to” policy for recipient based on SPF and DKIM
  - “fix” the alignment problem of SPF and DKIM

Sender Policy Framework (SPF)



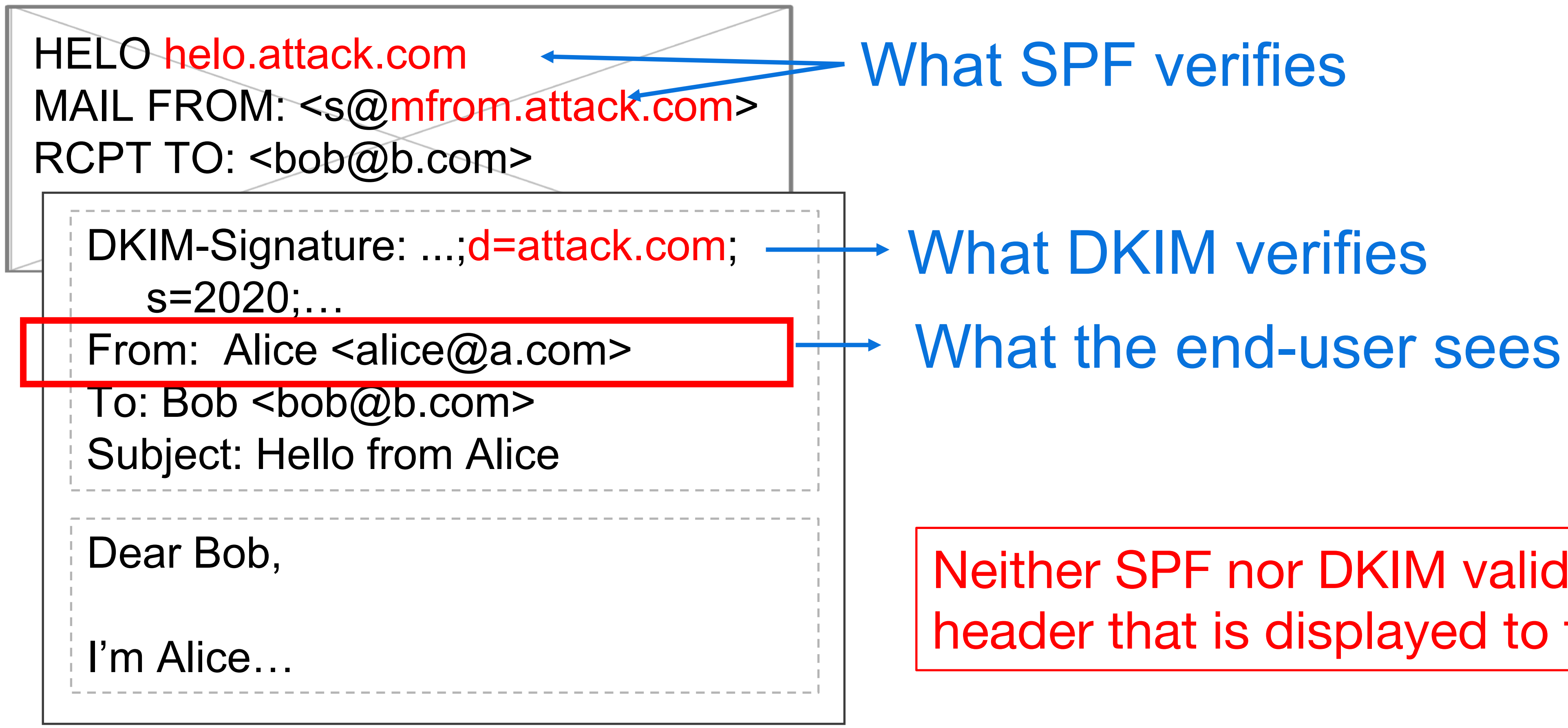
2 Query the domain in HELO and MAIL FROM to obtain the IP lists
HELO a.com
MAIL FROM: <s@a.com>
RCPT TO: <bob@b.com>
3 Check if the sender's IP matches the IP lists
• If yes, SPF pass
1 Publish authorized IP lists via DNS
From: Alice <alice@a.com>
To: Bob <bob@b.com>
Subject: Hello from Alice
Dear Bob,
I'm Alice…
DNS
DNS
a.com TXT 1.2.3.0/24
a.com TXT 1.2.3.0/24
MUA
Sending Services
1.2.3.4
Receiving Services
MUA
Alice
a.com
b.com
Bob
1.2.3.4 matches 1.2.3.0/24 ✓

DomainKeys Identified Mail (DKIM)



1 Publish public key via DNS
2 Generate DKIM-Signature with private key and attach it to the message.
3 Query "s._domainkey.d" (any._domainkey.a.com) to obtain public key
4 Validate DKIM signature with the public key
HELO ehlo.a.com
MAIL FROM: <s@mfrom.a.com>
RCPT TO: <bob@b.com>
DKIM-Signature: ...;d=a.com; s=any;…
From: Alice <alice@a.com>
To: Bob <bob@b.com>
Dear Bob,
I'm Alice…
DNS
DNS
MUA
Sending Services
Receiving Services
MUA
Alice
a.com
b.com
Bob

What’s Wrong with SPF/DKIM?



HELO helo.attack.com
MAIL FROM: <s@mfrom.attack.com>
RCPT TO: <bob@b.com>

DKIM-Signature: ...;d=attack.com;
    s=2020;…
From: Alice <alice@a.com>
To: Bob <bob@b.com>
Subject: Hello from Alice

Dear Bob,

I’m Alice…

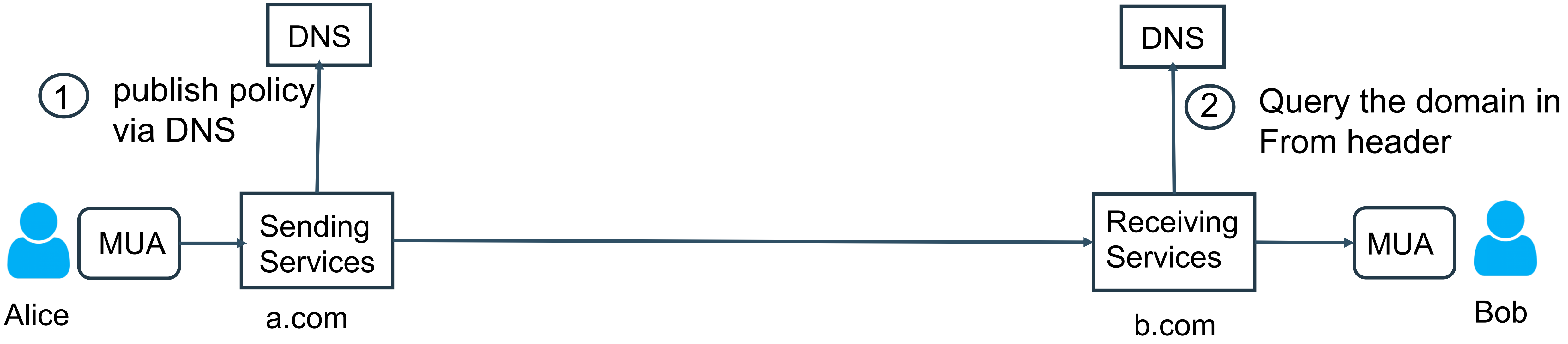
Neither SPF nor DKIM validate the From header that is displayed to the end-user.

Domain Message Authentication, Reporting and Conformance (DMARC)

③ Receiving services perform **identifier alignment test** to check if the domain in From header matches SPF or DKIM-verified domain.

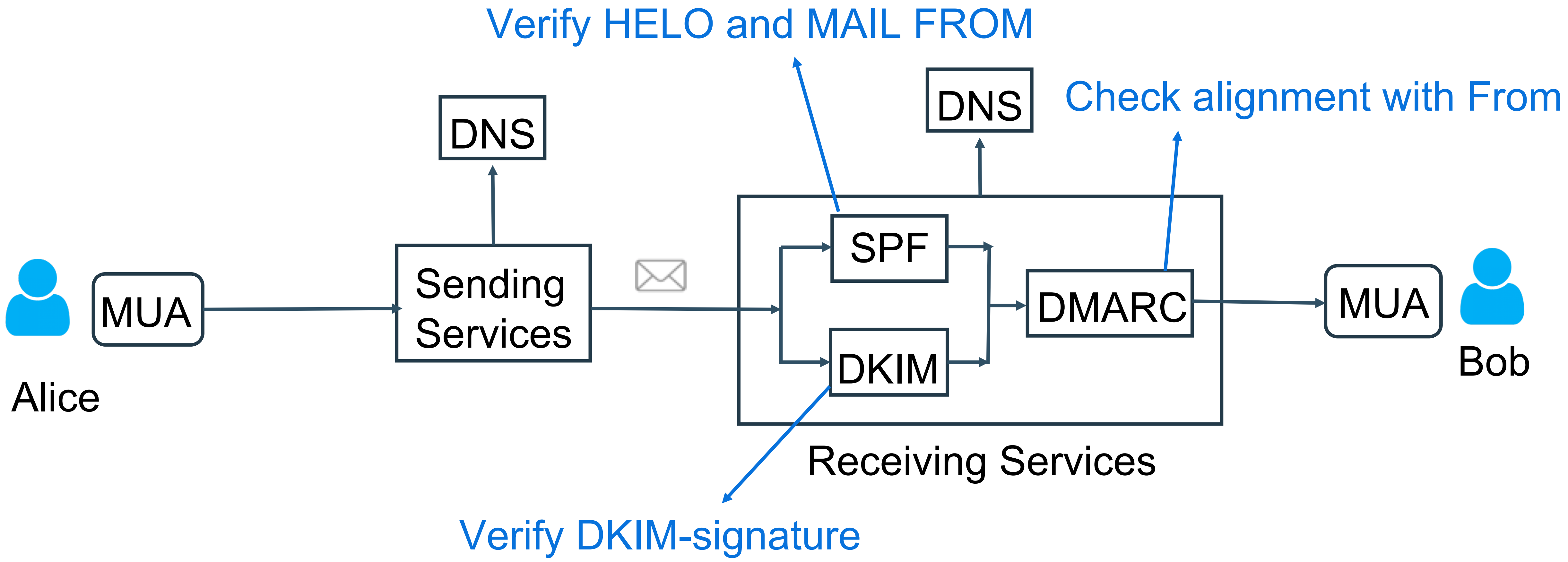
- Exactly match (strict) or have the same registered domain* (relaxed, default mode)

④ The email passes DMARC authentication if:
1) either SPF or DKIM show a positive result, and
2) the From header domain passes the alignment test.



* Defined in public suffix list, https://publicsuffix.org/

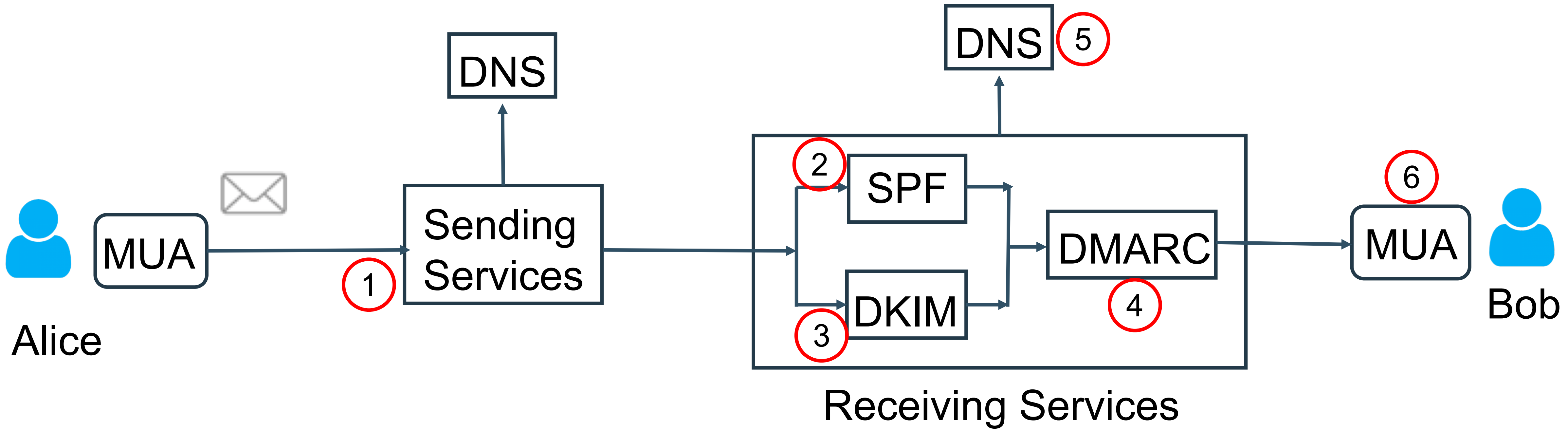
Overview of Email Authentication Flow



What could possibly go wrong?

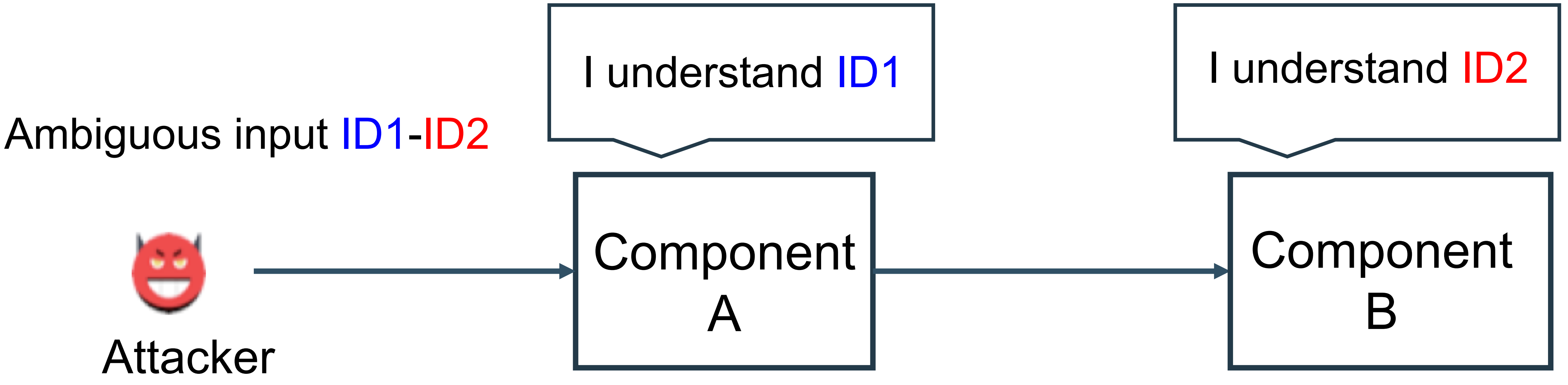
Bypassing the Authentication

Key Idea of Our Attacks



Inconsistencies between different components could lead to security vulnerabilities.

Key Idea of Our Attacks



Inconsistencies between different components could lead to security vulnerabilities.

Exp. 1: Inconsistencies b/w SPF and DMARC

SMTP defines multiple identifiers

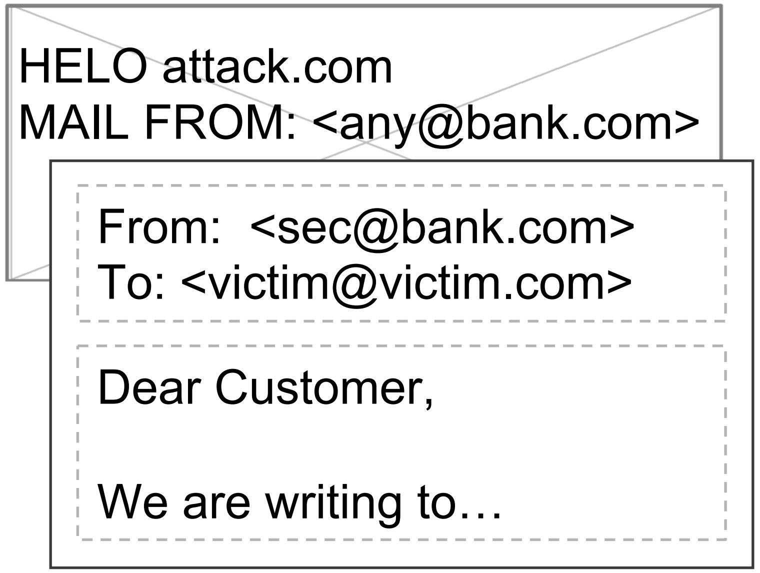
- HELO and MAIL FROM

SPF (RFC 7208)

- Check both HELO and MAIL FROM
- If either **fails**, SPF fails

DMARC (RFC 7489)

- Use MAIL FROM for alignment test.
- If MAIL FROM is empty, use HELO

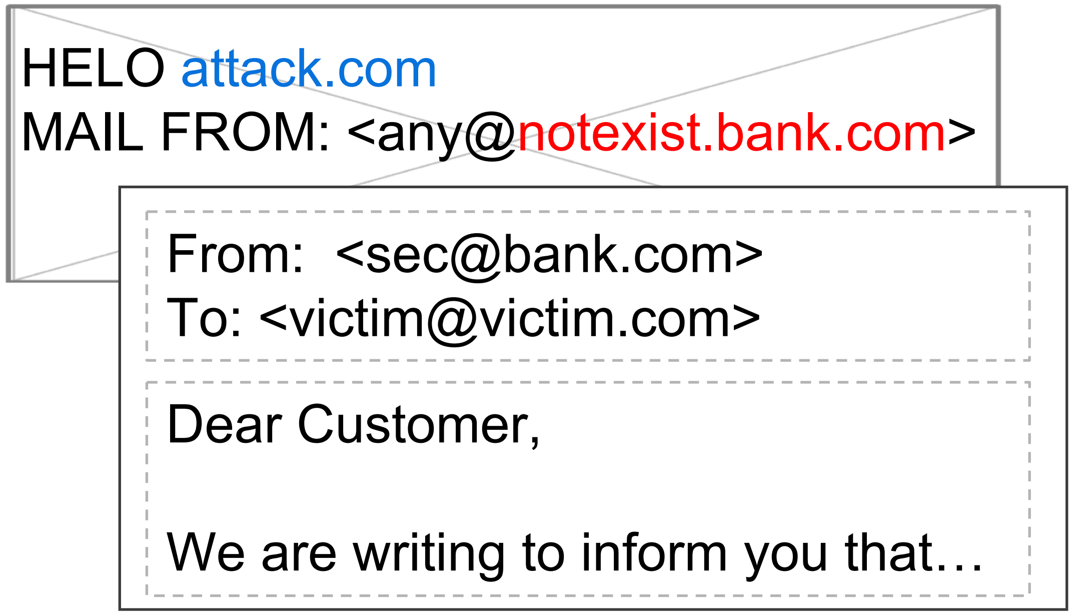


```
HELO attack.com
MAIL FROM: <any@bank.com>

From:  <sec@bank.com>
To: <victim@victim.com>

Dear Customer,

We are writing to…
```

Ambiguity: SPF uses **HELO**, and DMARC uses **MAIL FROM**

Exp. 1: Inconsistencies b/w SPF and DMARC

Ambiguity: SPF uses HELO, and DMARC uses MAIL FROM



```
HELO attack.com
MAIL FROM: <any@notexist.bank.com>
```

```
From:  <sec@bank.com>
To: <victim@victim.com>
```

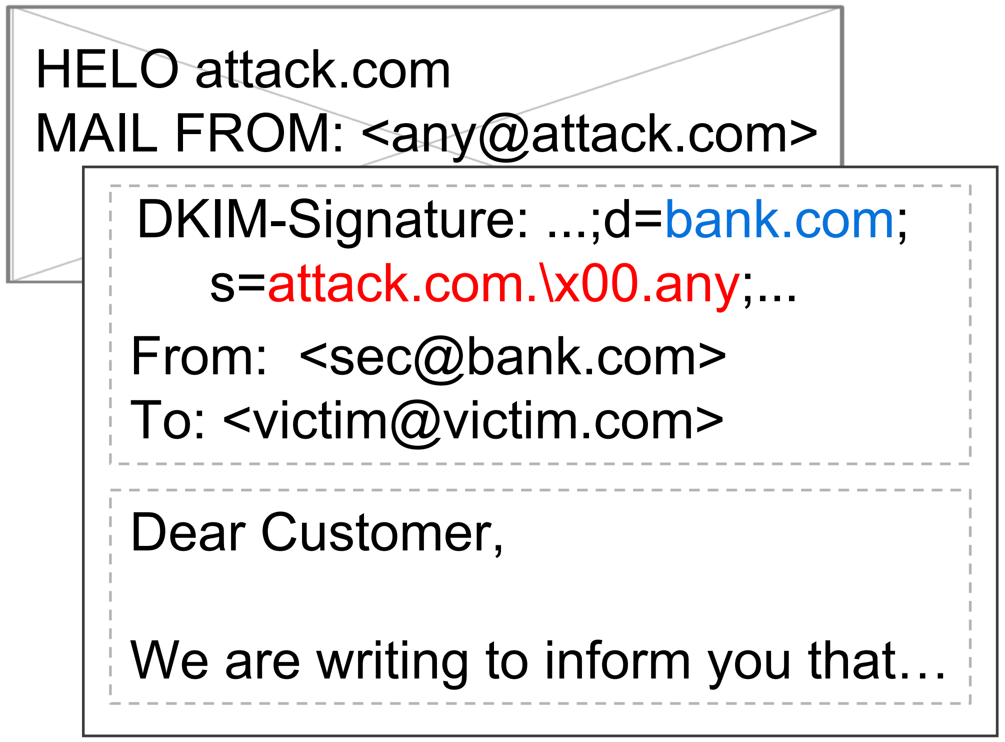
Dear Customer,

We are writing to inform you that…

1. SPF cannot verify MAIL FROM, and can only verify HELO
   - the non-existent domain doesn't have SPF policy, yet not considered as FAIL
2. DMARC uses MAIL FROM
   - because MAIL FROM is not empty
3. SPF **pass**, DMARC **pass**

Exp. 2: Inconsistencies b/w DKIM and DNS

Ambiguity: What DKIM uses differs from what DNS queries



```
HELO attack.com
MAIL FROM: <any@attack.com>

DKIM-Signature: ...;d=bank.com;
     s=attack.com.\x00.any;...
From:  <sec@bank.com>
To: <victim@victim.com>

Dear Customer,

We are writing to inform you that…
```

1. Attacker signs the message with his private key and sends the message
2. When receiving the message, DKIM use 'attack.com.\x00.any._domainkey.bank.com' to obtain the public key
3. But DNS takes \x00 as a terminator, and obtains public key from *attack.com*
4. DKIM **pass**, DMARC **pass**

Exp. 3: Authentication Results Injection

Ambiguity: Exploiting how SPF/DKIM forwards results to DMARC

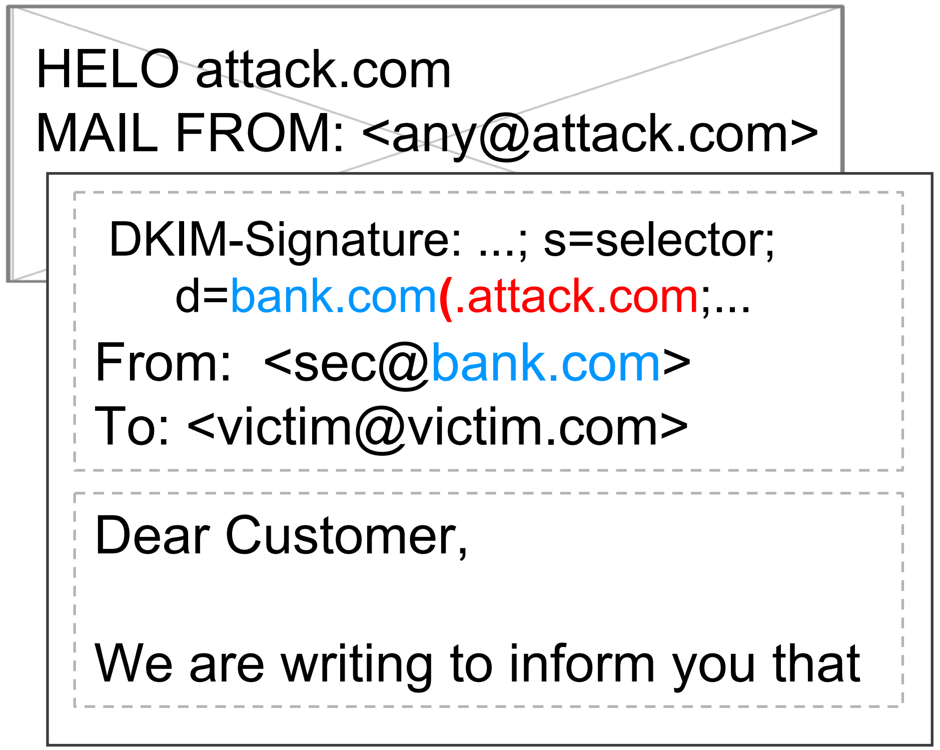
RFC 8601 define Authentication-Results header for communicating results between SPF/DKIM and DMARC :

Comments

```
Authentication-Results: example.com; spf=pass
    smtp.mailfrom=sender@sender.com; dkim=pass (1024-bit key)
    reason="signature ok" header.d=sender.com;
```

DMARC extracts “smtp.mailfrom” and “header.d” to check alignment with From header.

Exp. 3a: DKIM Authentication Results Injection



```
HELO attack.com
MAIL FROM: <any@attack.com>

DKIM-Signature: ...; s=selector;
    d=bank.com(.attack.com;...
From:  <sec@bank.com>
To: <victim@victim.com>

Dear Customer,

We are writing to inform you that
```

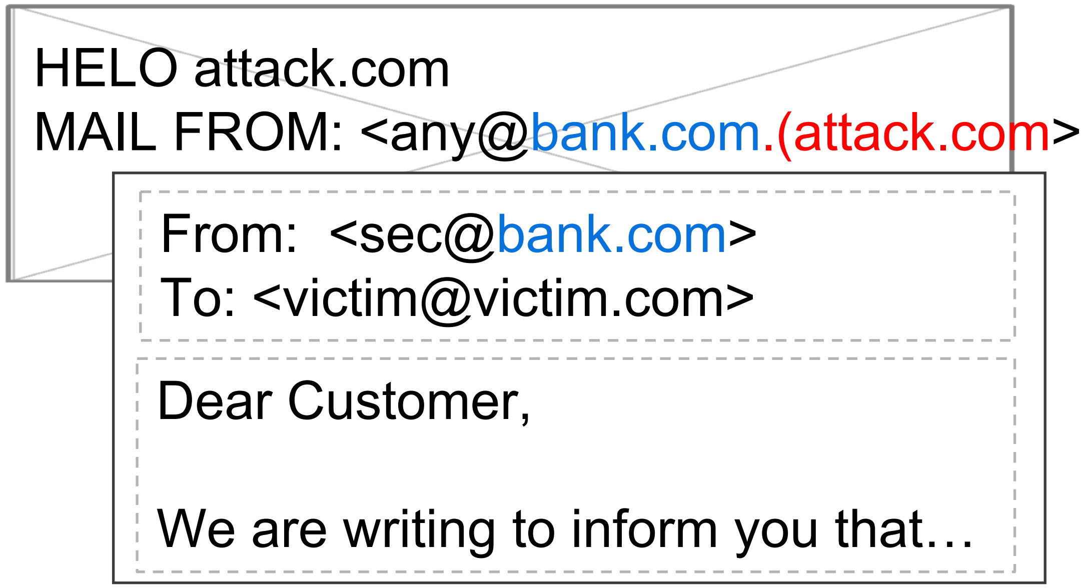
1. Attacker signs the message with their private key
2. DKIM verifies the message with the attacker's public key from 'selector._domainkey.bank.com(.attack.com' and generates:

```
Authentication-results: bank.com;
    dkim=pass (1024-bit key)
    header.d=bank.com(.attack.com;
```

Comments

3. DMARC parses the content after the "(" as a comment, and uses bank.com to check alignment with From header
4. DKIM **pass**, DMARC **pass**

Exp. 3b: SPF Authentication Results Injection



HELO attack.com
MAIL FROM: <any@bank.com.(attack.com>

From: <sec@bank.com>
To: <victim@victim.com>

Dear Customer,

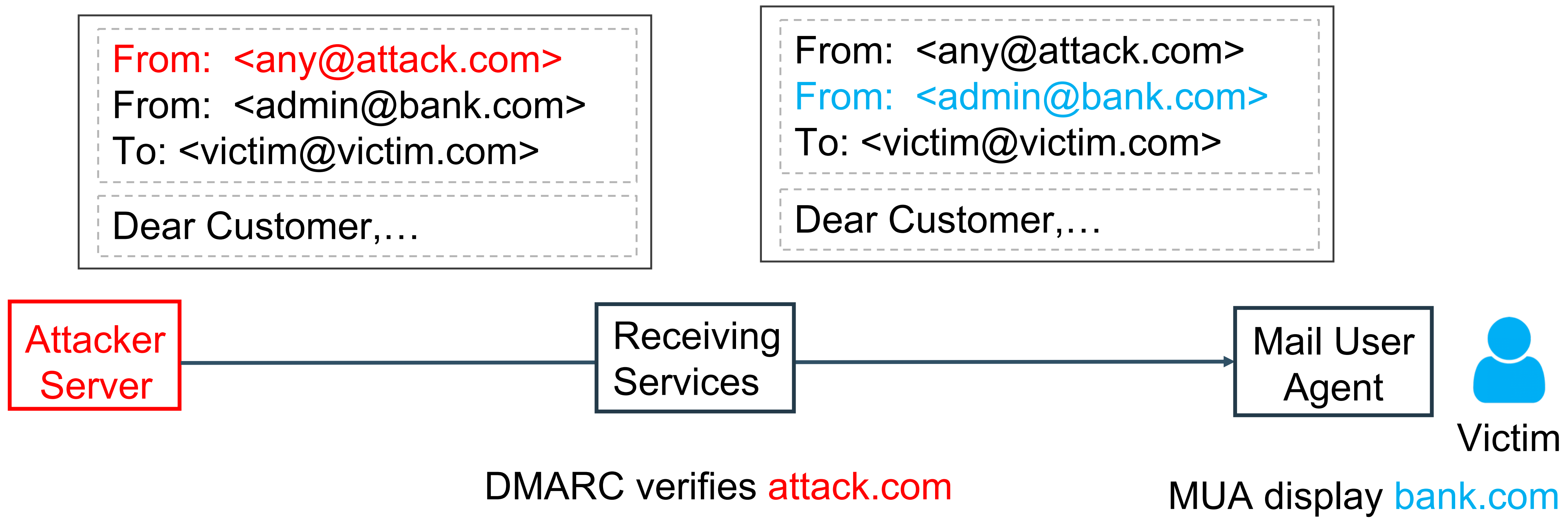
We are writing to inform you that…

- SPF verifies bank.com(.attack.com
- DMARC uses bank.com to check alignment with From header
- SPF **pass**, DMARC **pass**

Attacker can also use single (‘) and double (“) quotes to replace “(“.

Exp. 4a: Multiple From Headers

Ambiguity: What receiving server verifies differ from what MUA displays

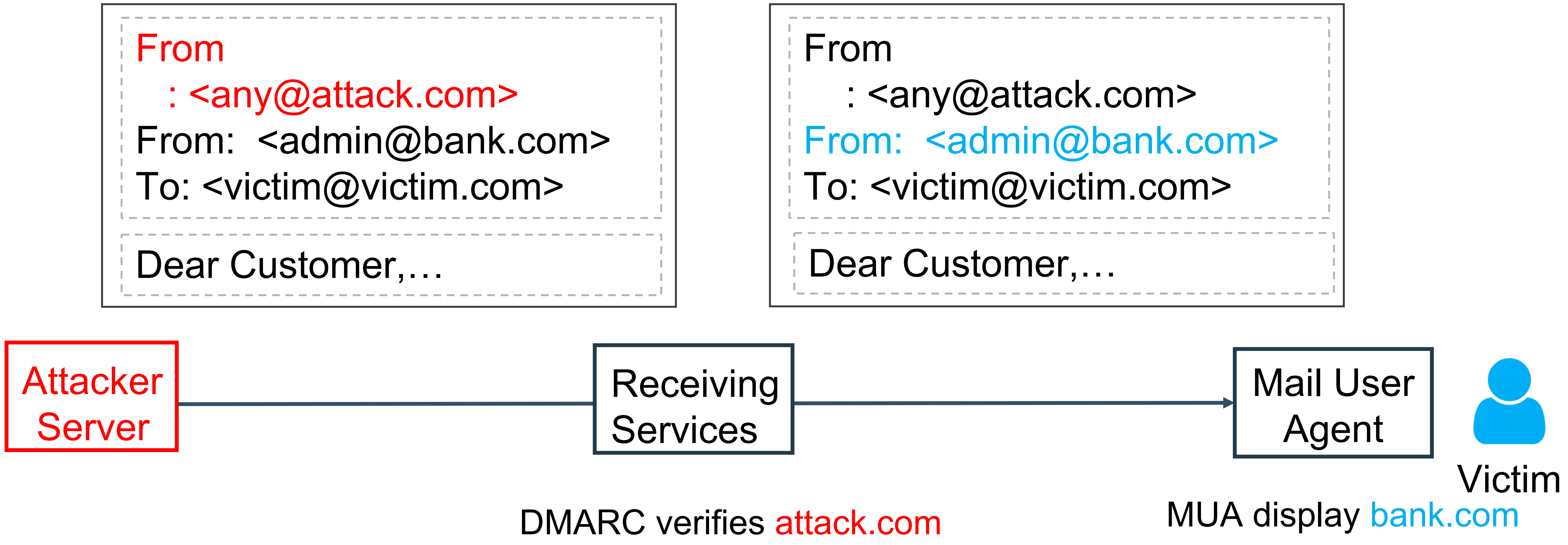


- RFC 5322: Messages with multiple From should be rejected
- In practice: 19/29 accept (15 use first, 3 use last, 1 show both)

Exp. 4b: Multiple From Headers with Space

Three types of variants:

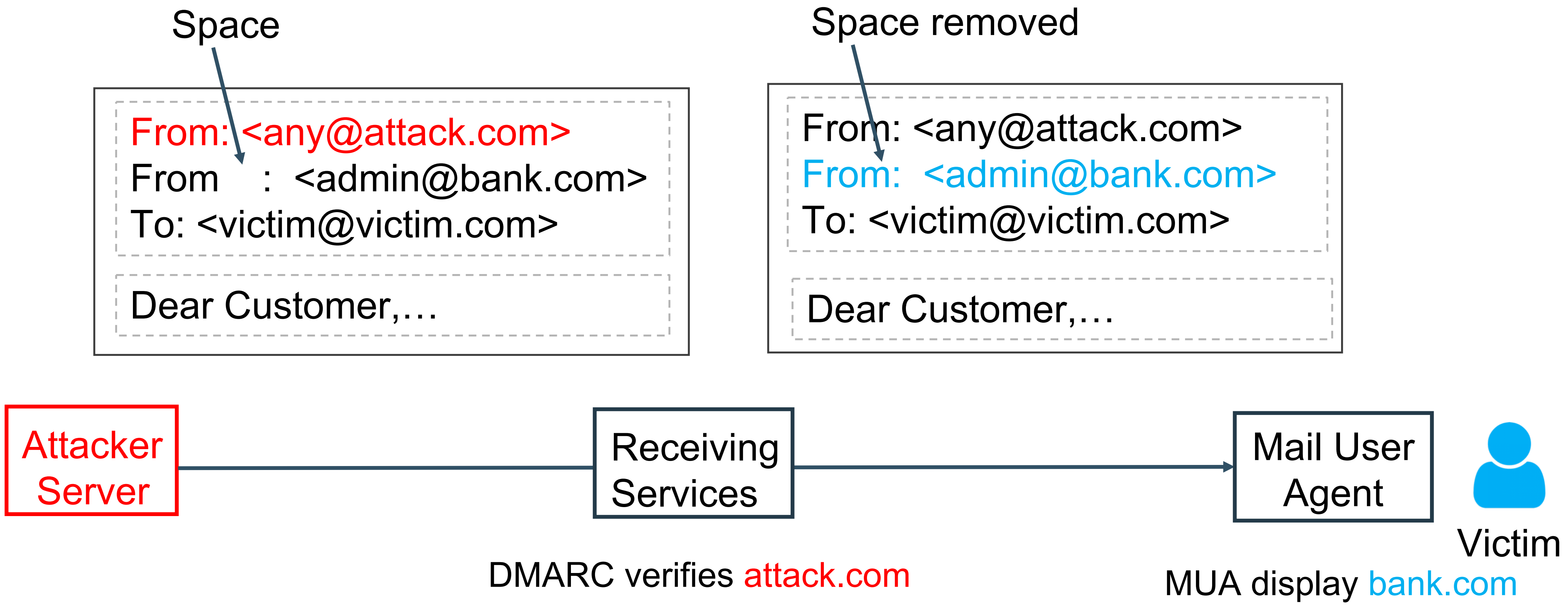
1) _From: a@a.com ; 2) From_: a@a.com; 3) From\r\n_: a@a.com



```
From
   : <any@attack.com>
From:  <admin@bank.com>
To: <victim@victim.com>

Dear Customer,…
```

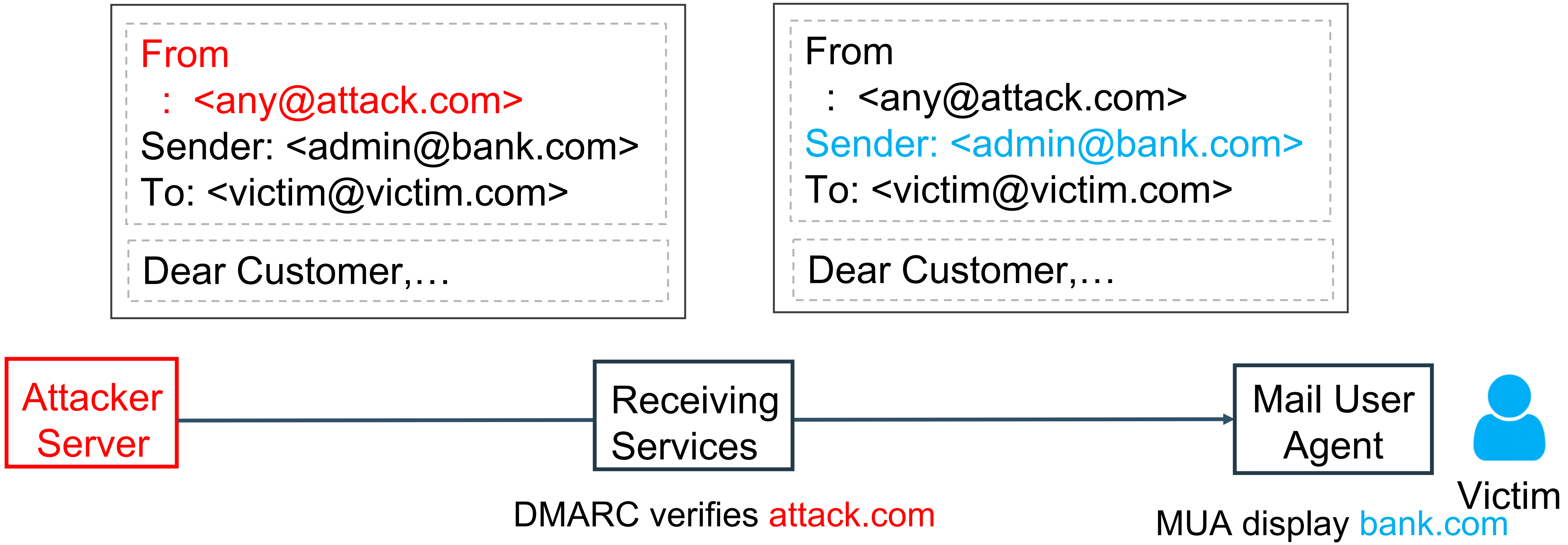
```
From
    : <any@attack.com>
From:  <admin@bank.com>
To: <victim@victim.com>

Dear Customer,…
```

Exp. 4c: Multiple From Headers with Normalization

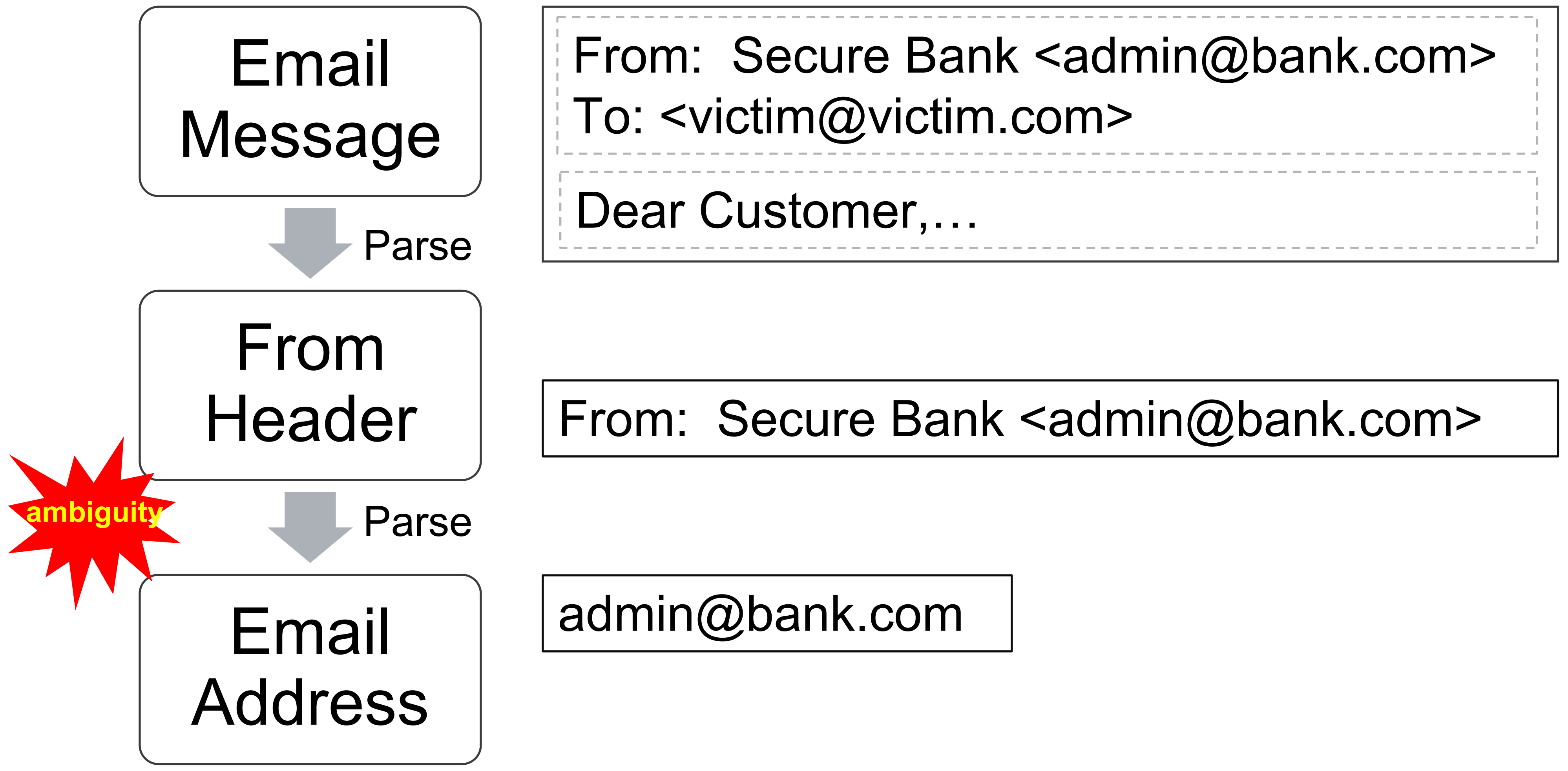


Exp. 5: From/Sender Ambiguity

- 7/19 MUAs display Sender or Resent-From header value when From header is absent



Email Parsing Process



Complex From Header Syntax

Display Name Comments Route portion Real address

```
From:  Secure (b@b.com) Bank <@c.com, @d.com:
a@a.com (e@e.com) > (f@f.com)
```

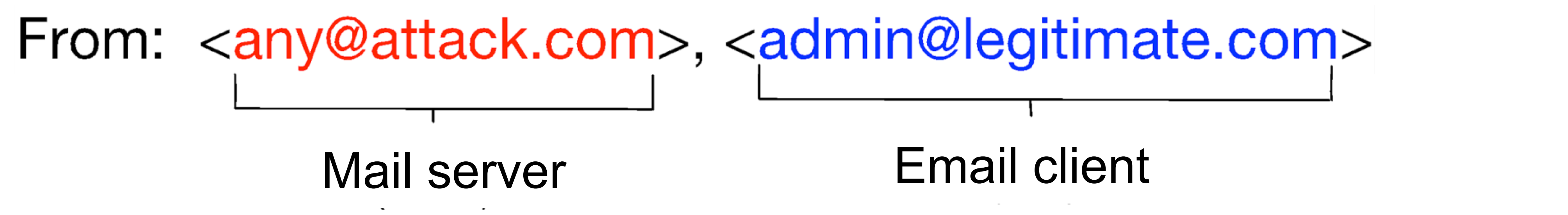
A quick example of valid (!) From header

- **Multiple address lists.** [RFC 5322]
- **Encoding**: defined to support no-ascii character. [RFC 2047]

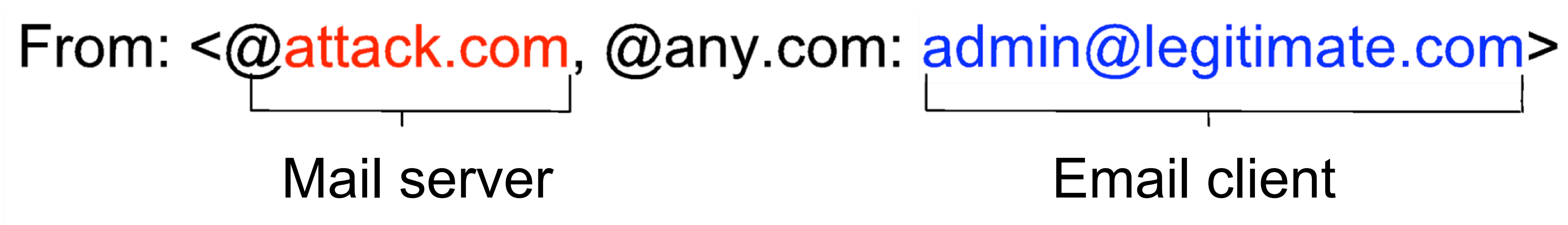
  From: bob <b@b.com> is equal to

  From: =?utf-8?B?Ym9i?=<b@b.com> in Base64 encoding
- **Quoted-pair**: use ‘\’ to escape special characters like ‘(‘. [RFC 5322]

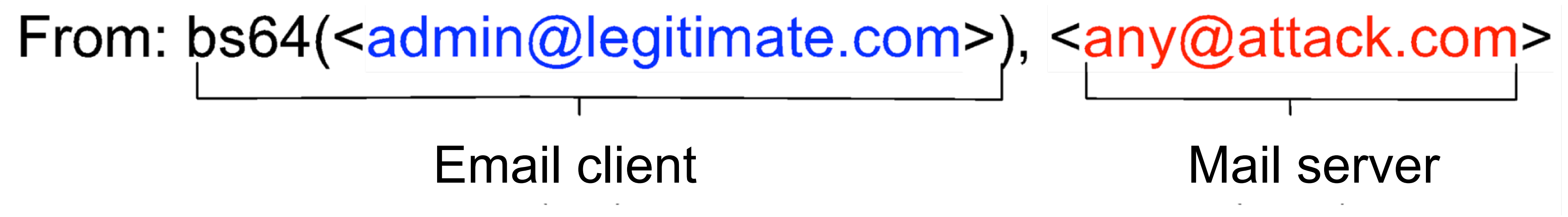
Exp. 6a: Exploiting Differences in Feature Support



```
From:  <any@attack.com>, <admin@legitimate.com>
```



```
From: <@attack.com, @any.com: admin@legitimate.com>
```



```
From: bs64(<admin@legitimate.com>), <any@attack.com>
```

Exp. 6b: Exploiting Parsing Inconsistencies

$$\text{From: <}\underbrace{\text{admin@legitimate.com}}_{\text{Email client}}\text{>}\backslash\text{, <}\underbrace{\text{any@attack.com}}_{\text{Mail server}}\text{>}$$

$$\text{From: }\underbrace{\text{admin@legitimate.com}}_{\text{Email client}}\text{, <}\underbrace{\text{any@attack.com}}_{\text{Mail server}}\text{>}$$

$$\text{From: <}\underbrace{\text{any@attack.com}}_{\text{Mail server}}\text{>}\underbrace{\text{admin@legitimate.com}}_{\text{Email client}}$$

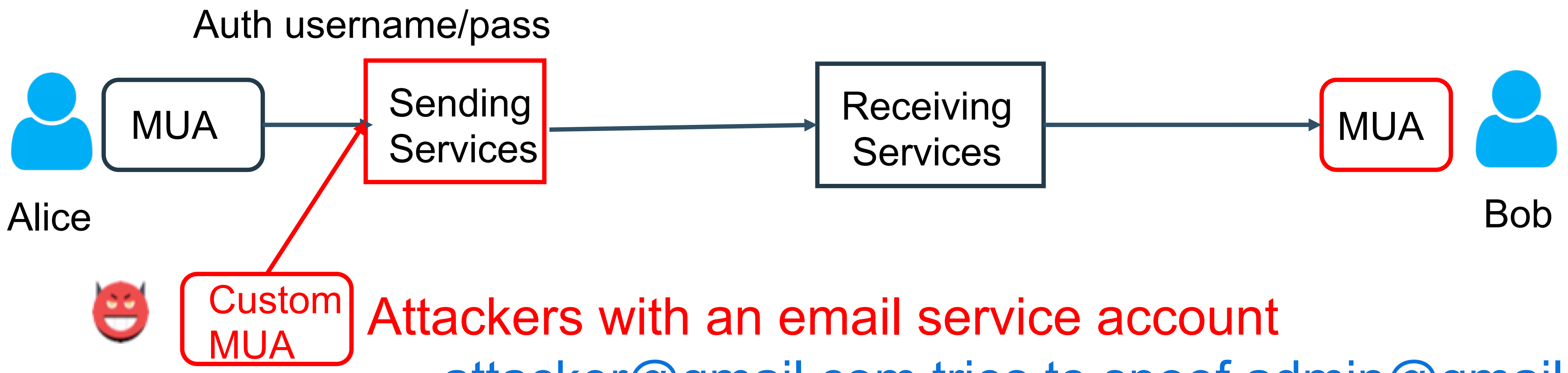
How Prevalent are UI-mismatch Vulnerabilities?

- We tested 10 popular email providers and 19 email clients
- 43 out of 82 different combinations that could be exploited
- What we found only constitutes a subset of the problem

Read our paper for more details

Exp. 7: Spoofing via an Email Service Account

Ambiguity: What sending server validates differ from what MUA displays



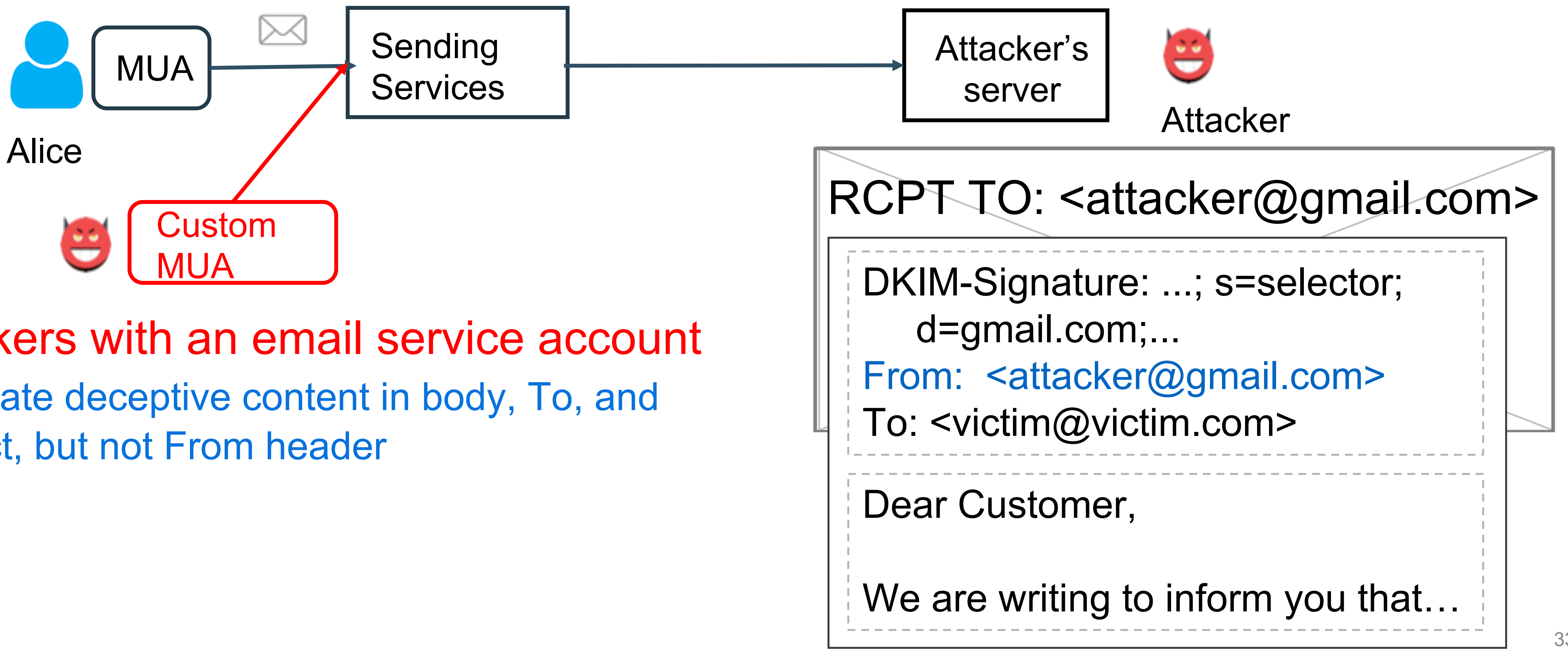
Attackers with an email service account

- attacker@gmail.com tries to spoof admin@gmail.com

- Sending services should ensure that the From header matches authenticated username
  - But From header validation is error-prone because of complex syntax
- We found 7 out of 8 email providers are vulnerable

Exp. 8: Combing Replay and Multiple-From Ambiguity (1/2)

① Attacker emails himself through the email provider server.



RCPT TO: <attacker@gmail.com>

DKIM-Signature: ...; s=selector;
d=gmail.com;...
From: <attacker@gmail.com>
To: <victim@victim.com>

Dear Customer,

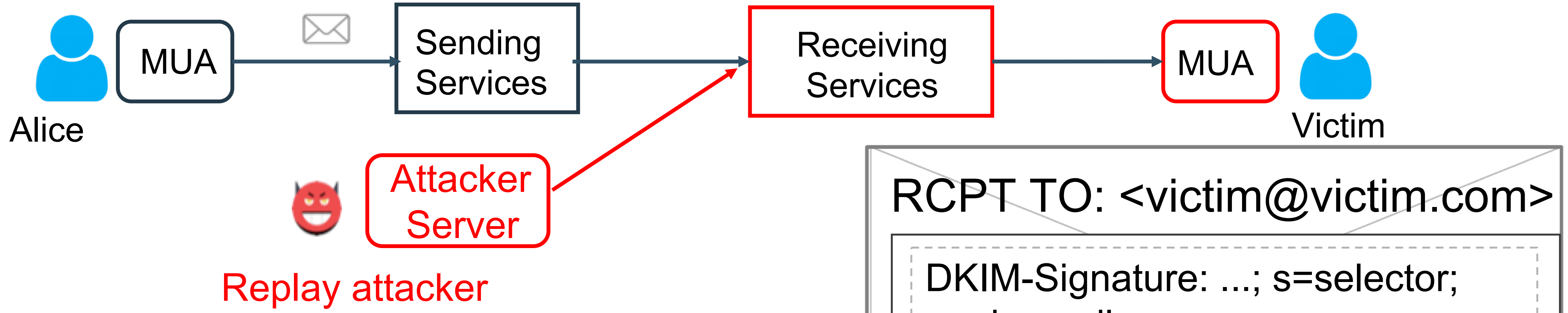
We are writing to inform you that…

Attackers with an email service account

- Create deceptive content in body, To, and Subject, but not From header

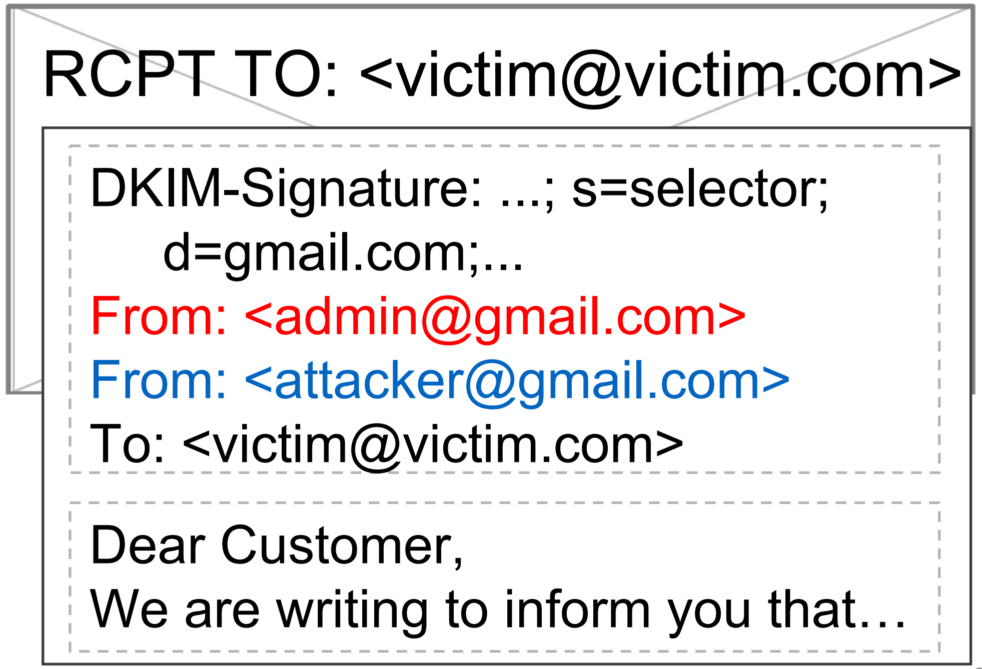
Exp. 8: Combing Replay and Multiple-From Ambiguity (2/2)

② Attacker replays the messages with an extra From header.



Ambiguity: What DKIM verifies differs from what MUA displays

- DKIM components verify the last header
- MUAs show the first header



RCPT TO: <victim@victim.com>

DKIM-Signature: ...; s=selector;
    d=gmail.com;...
From: <admin@gmail.com>
From: <attacker@gmail.com>
To: <victim@victim.com>

Dear Customer,
We are writing to inform you that…

Thinking on Defense

- Better parsing and protocol spec
  - "Be ~~liberal~~ **strict** in what you accept"
  - make protocol implementation-friendly
    - simple, well-typed/structured messages, reduce/avoid multiple party processing
- Better UI
  - UI needs more explicit security indicators
- For end-users
  - Don't blindly trust the email sender displayed in email client
  - Use end-to-end authentication such as PGP
    - PGP may also have parsing ambiguities, but hopefully better than those in SPF/DKIM/DMARC.

New tool - espoofer

We will make this tool publicly available at

https://github.com/chenjj/espoofer

Thank you!

See more demo videos at here, full paper at here.