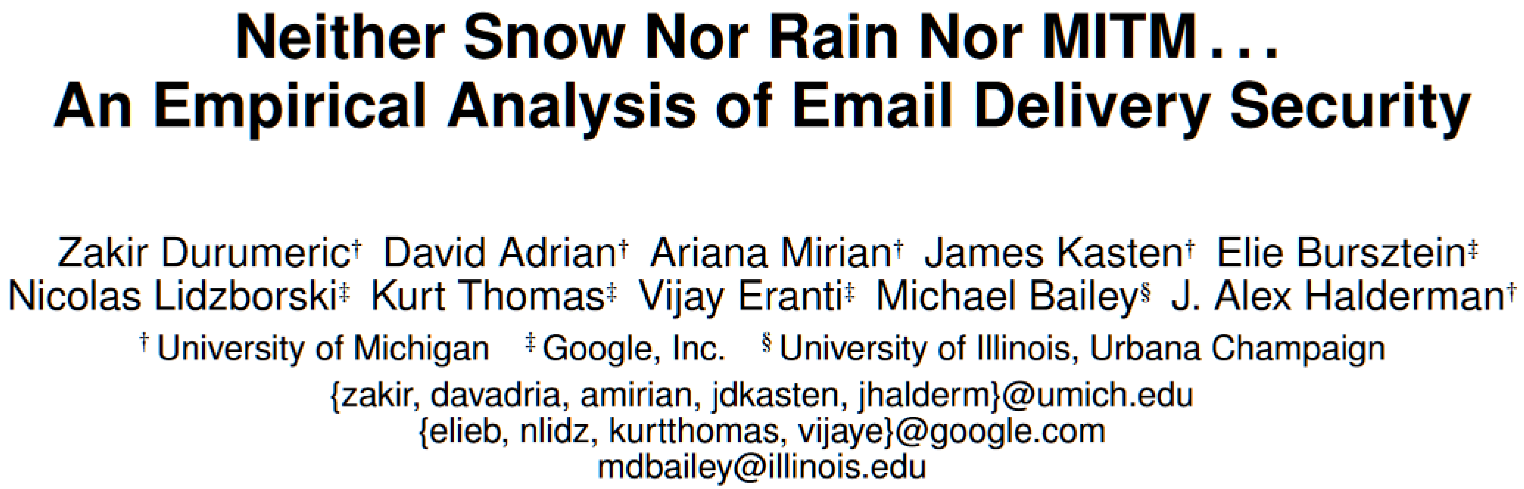
Email Security

CS558 Network Security

Boston University

Prof. S. Goldberg, Spring 2017



Neither Snow Nor Rain Nor MITM . . . An Empirical Analysis of Email Delivery Security

Zakir Durumeric[†] David Adrian[†] Ariana Mirian[†] James Kasten[†] Elie Bursztein[‡]
Nicolas Lidzborski[‡] Kurt Thomas[‡] Vijay Eranti[‡] Michael Bailey[§] J. Alex Halderman[†]
[†] University of Michigan [‡] Google, Inc. [§] University of Illinois, Urbana Champaign
{zakir, davadria, amirian, jdkasten, jhalderm}@umich.edu
{elieb, nlidz, kurtthomas, vijaye}@google.com
mdbailey@illinois.edu

https://www.elie.net/blog/understanding-how-tls-downgrade-attacks-prevent-email-encryption

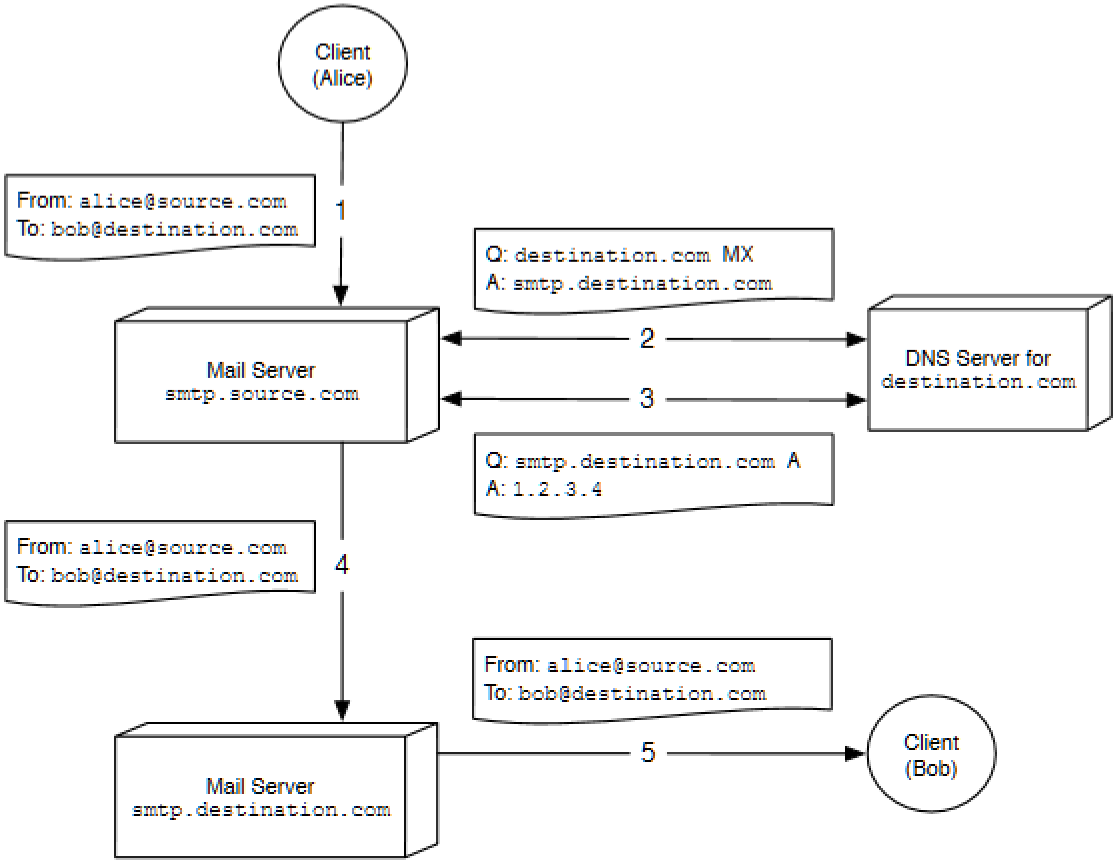
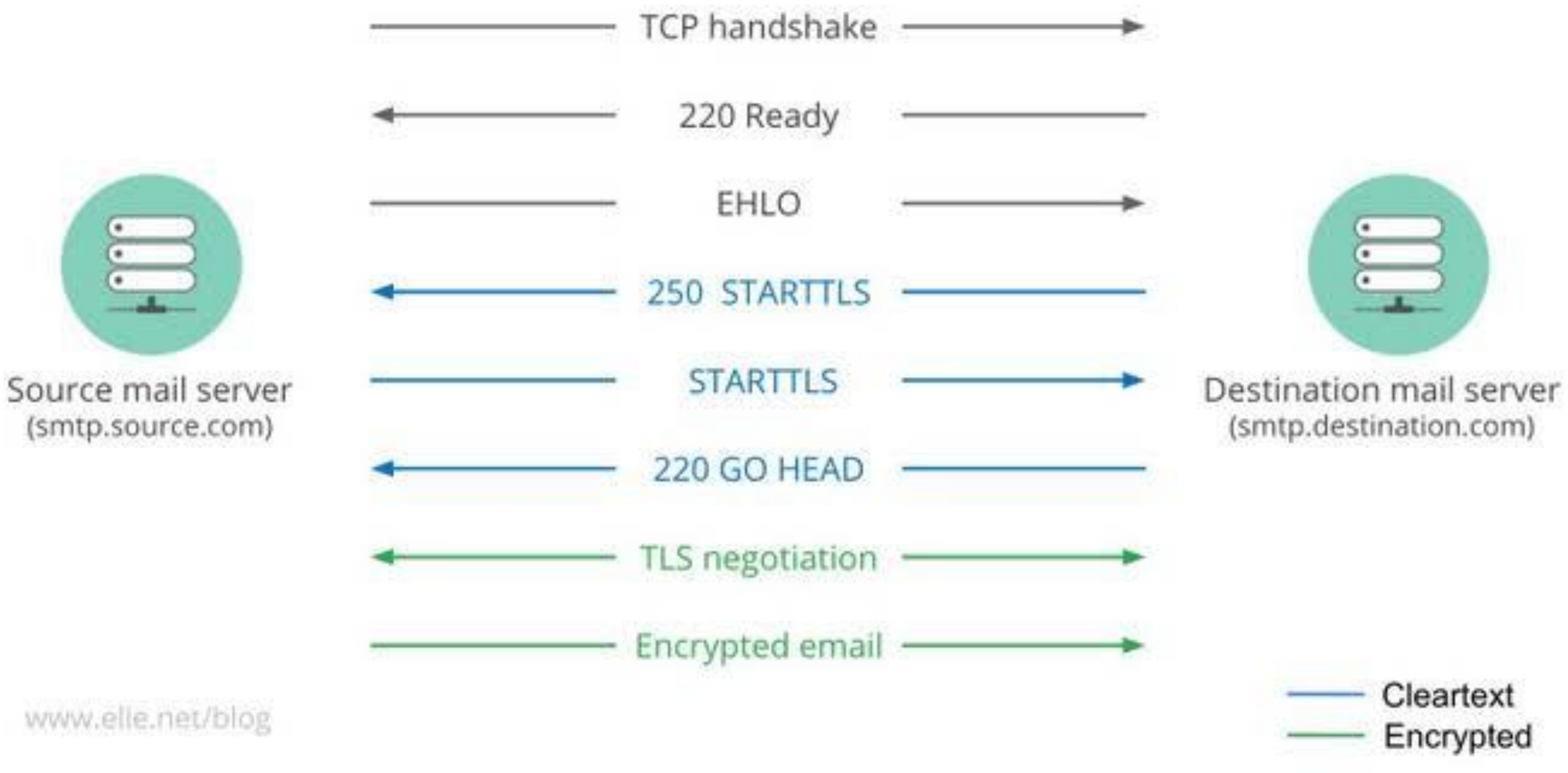


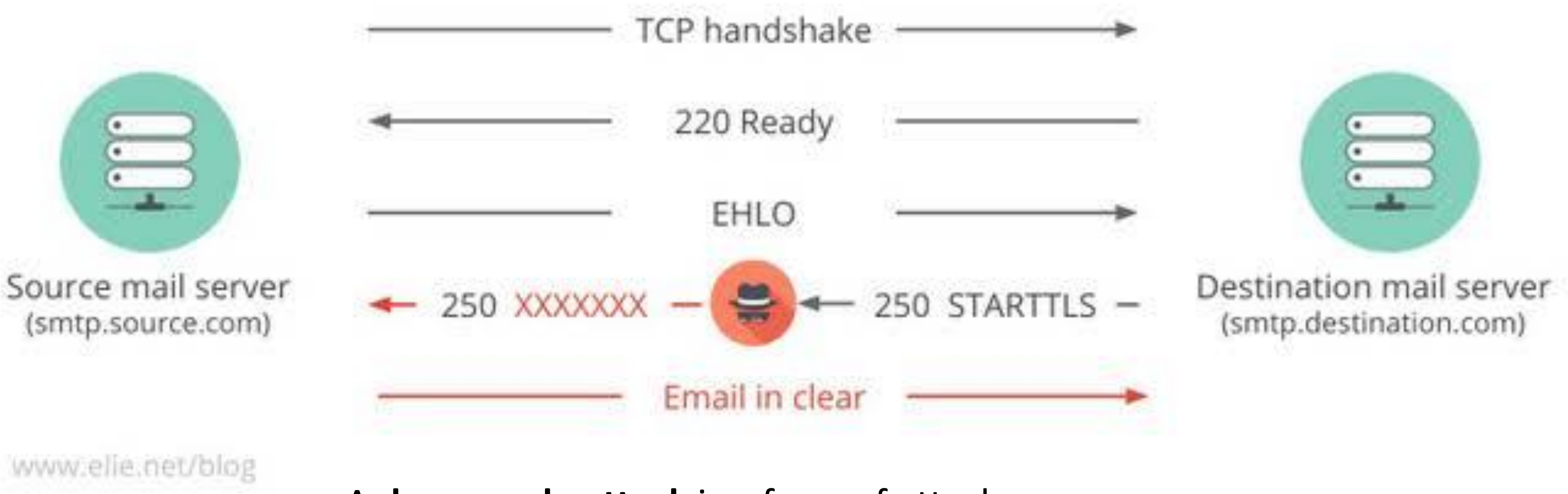
Figure 1: **SMTP Protocol**—A client sends outgoing mail by connecting to its organization's local SMTP server (❶). The local server performs a DNS lookup for the mail exchange (MX) record of the *destination.com* domain, which contains the hostname of the destination's SMTP server, in this case *smtp.destination.com* (❷). The sender's server then performs a second DNS lookup for the destination server's IP address (❸), establishes a connection, and relays the message (❹). The recipient can later retrieve the message using a secondary protocol such as POP3 or IMAP (❺).

STARTTLS



As visible in the diagram above, this retrofitting was achieved by adding the verb STARTTLS to the SMTP options command that the server sends to the SMTP client as part as the protocol negotiation. If the client supports encryption (TLS), it will understand the STARTTLS verb and will initiate a TLS exchange before sending the email to ensure it is encrypted. If the client doesn’t know TLS, it will simply ignore the STARTTLS and send the email in clear

STARTTLS Downgrade Attack



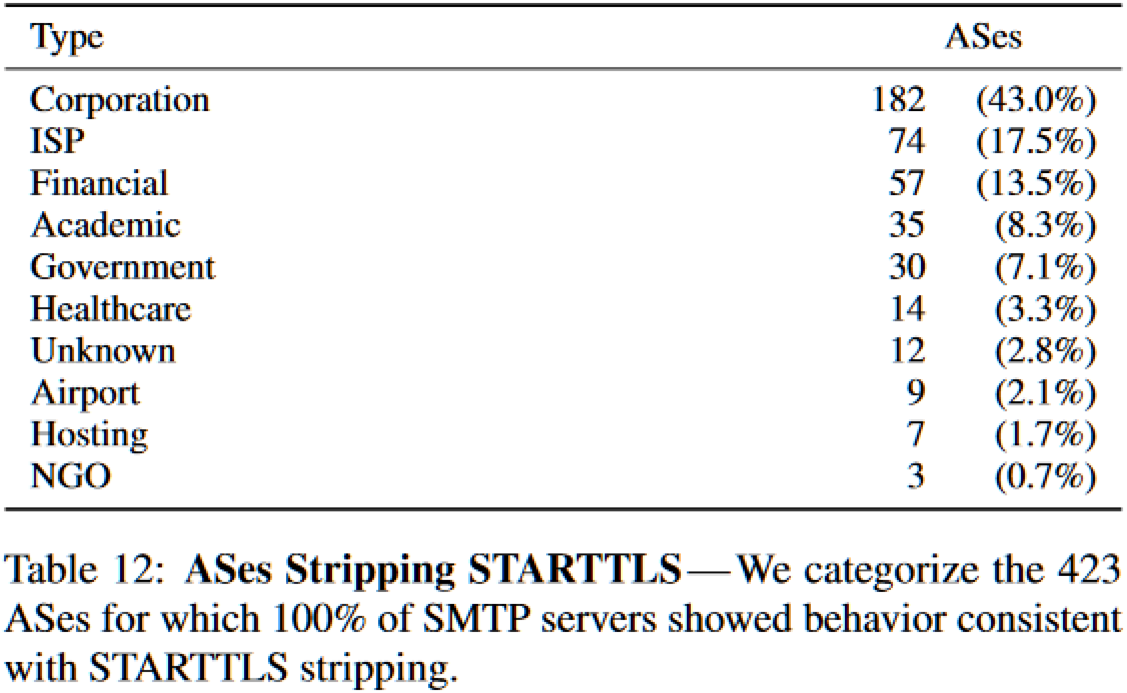
A **downgrade attack** is a form of attack on a computer system or communications protocol that makes it abandon a high-quality mode of operation (e.g. an encrypted connection) in favor of an old, lower-quality mode of operation (e.g. clear text) that is there for backward compatibility with older systems.

Mail Software	Top Million Market Share	Public IPv4 Market Share	STARTTLS Incoming	STARTTLS Outgoing	Server Validation	Domain Validation	Reject Invalid Certificates	TLS Version
exim 4.82	34%	24%	◐	●	○	○	○	1.2
Postfix 2.11.0	18%	21%	●	◐	◐	◐	◐	1.2
qmail 1.06	6%	1%	◐	◐	○	○	○	1.2
sendmail 8.14.4	5%	4%	◐	●	○	○	○	1.2
Exchange 2013	4%	12%	●	●	◐	○	◐	1.0
Other	3%	<1%						
Unknown	30%	38%	● default behavior | ◐ supported but not default | ○ no support					

Table 6: **Popular Mail Transfer Agents (MTA)**—We investigated the default behavior for five popular MTAs. By default, Postfix and qmail do not initiate STARTTLS connections. All five MTAs we tested fail open to cleartext if the STARTTLS connection fails.

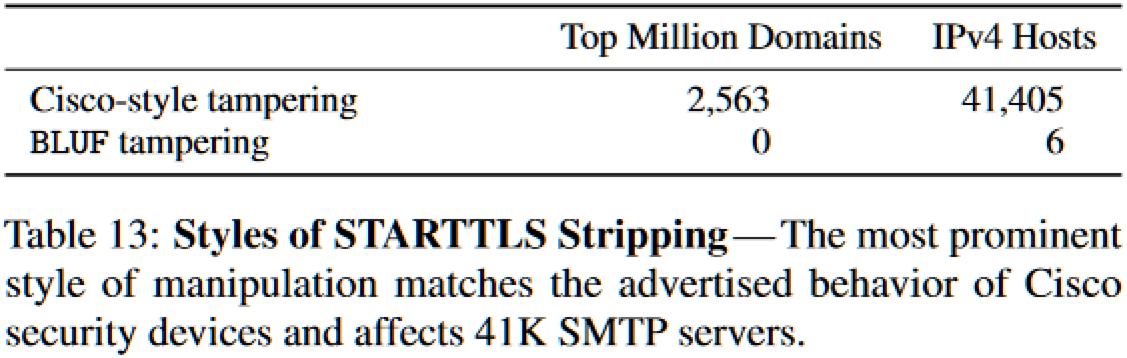
In order to understand why such a large number of organizations have not deployed STARTTLS and why only half of inbound connections to Gmail initiate a STARTTLS connection, we investigated the five most popular SMTP implementations, which account for 97% of identifiable mail servers for the Top Million domains. We tested whether each implementation initiated STARTTLS connections, whether it supported STARTTLS for incoming connections, and how it validated certificates. We installed the latest version of each SMTP server on an Ubuntu 14.04.1 LTS system, except for Microsoft Exchange, which was readily documented online [37]. The results are summarized in Table 6.

STARTTLS Stripping



Type	ASes	
Corporation	182	(43.0%)
ISP	74	(17.5%)
Financial	57	(13.5%)
Academic	35	(8.3%)
Government	30	(7.1%)
Healthcare	14	(3.3%)
Unknown	12	(2.8%)
Airport	9	(2.1%)
Hosting	7	(1.7%)
NGO	3	(0.7%)

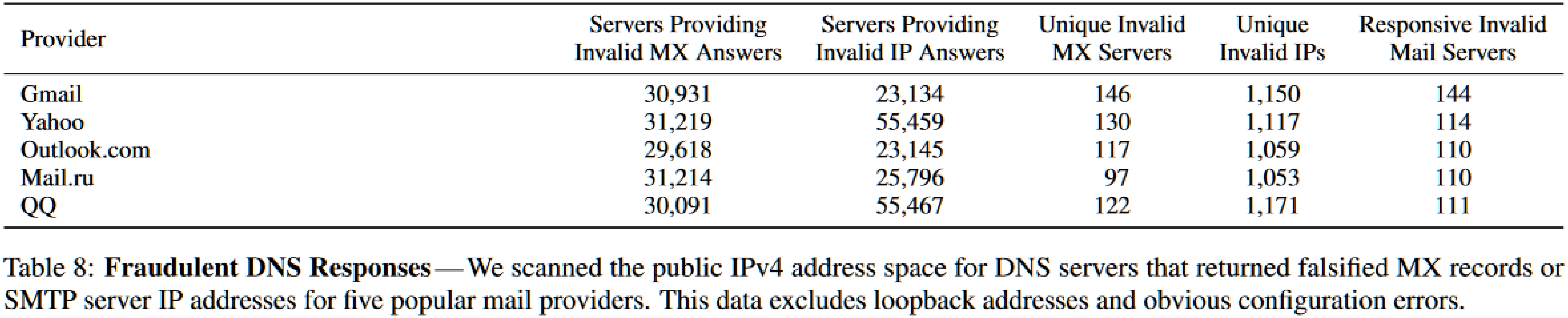
Table 12: **ASes Stripping STARTTLS**—We categorize the 423 ASes for which 100% of SMTP servers showed behavior consistent with STARTTLS stripping.



	Top Million Domains	IPv4 Hosts
Cisco-style tampering	2,563	41,405
BLUF tampering	0	6

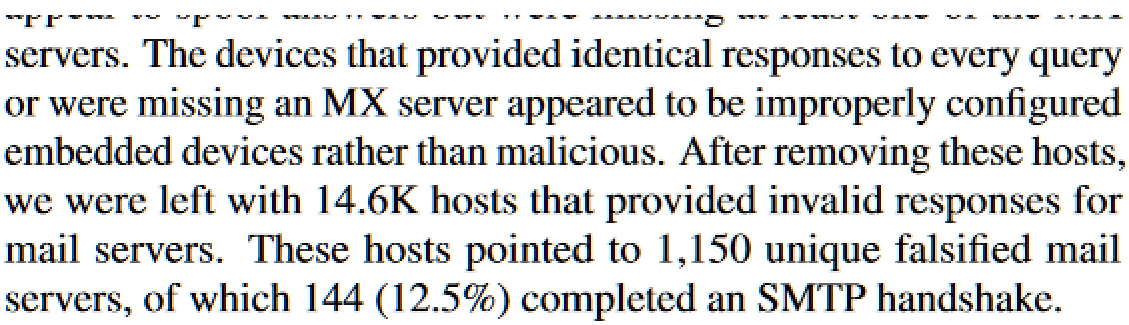
Table 13: **Styles of STARTTLS Stripping**—The most prominent style of manipulation matches the advertised behavior of Cisco security devices and affects 41K SMTP servers.

Fraudulent DNS responses



Provider	Servers Providing Invalid MX Answers	Servers Providing Invalid IP Answers	Unique Invalid MX Servers	Unique Invalid IPs	Responsive Invalid Mail Servers
Gmail	30,931	23,134	146	1,150	144
Yahoo	31,219	55,459	130	1,117	114
Outlook.com	29,618	23,145	117	1,059	110
Mail.ru	31,214	25,796	97	1,053	110
QQ	30,091	55,467	122	1,171	111

Table 8: **Fraudulent DNS Responses**—We scanned the public IPv4 address space for DNS servers that returned falsified MX records or SMTP server IP addresses for five popular mail providers. This data excludes loopback addresses and obvious configuration errors.



servers. The devices that provided identical responses to every query or were missing an MX server appeared to be improperly configured embedded devices rather than malicious. After removing these hosts, we were left with 14.6K hosts that provided invalid responses for mail servers. These hosts pointed to 1,150 unique falsified mail servers, of which 144 (12.5%) completed an SMTP handshake.

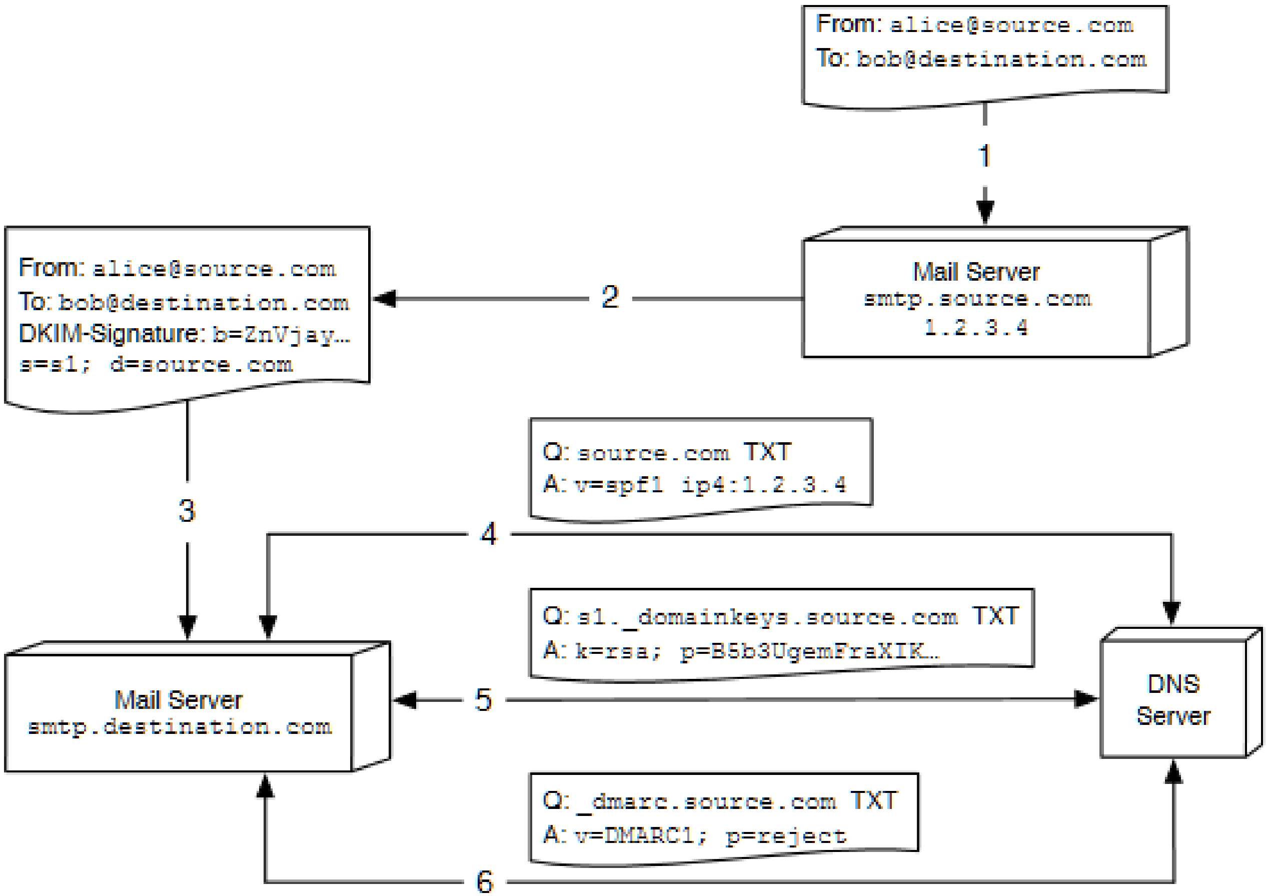
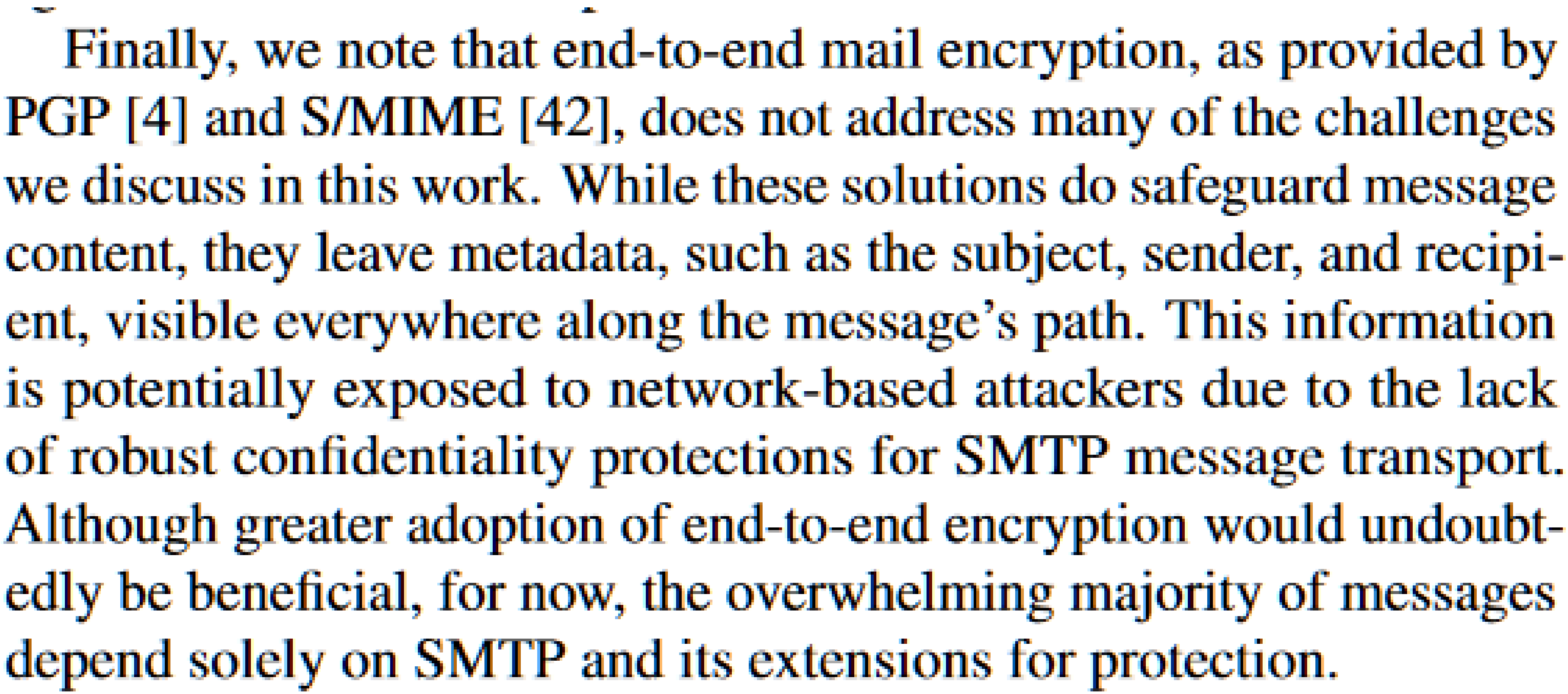


Figure 2: **Mail Authentication**—SPF, DKIM, and DMARC are used to provide source authentication. The outgoing server digitally signs the message (❷). The receiving mail server performs an SPF lookup (❹) to check if the outgoing server is whitelisted, a DKIM lookup (❺) to determine the public key used in the signature, and a DMARC lookup (❻) to determine the correct action should SPF or DKIM validation fail.

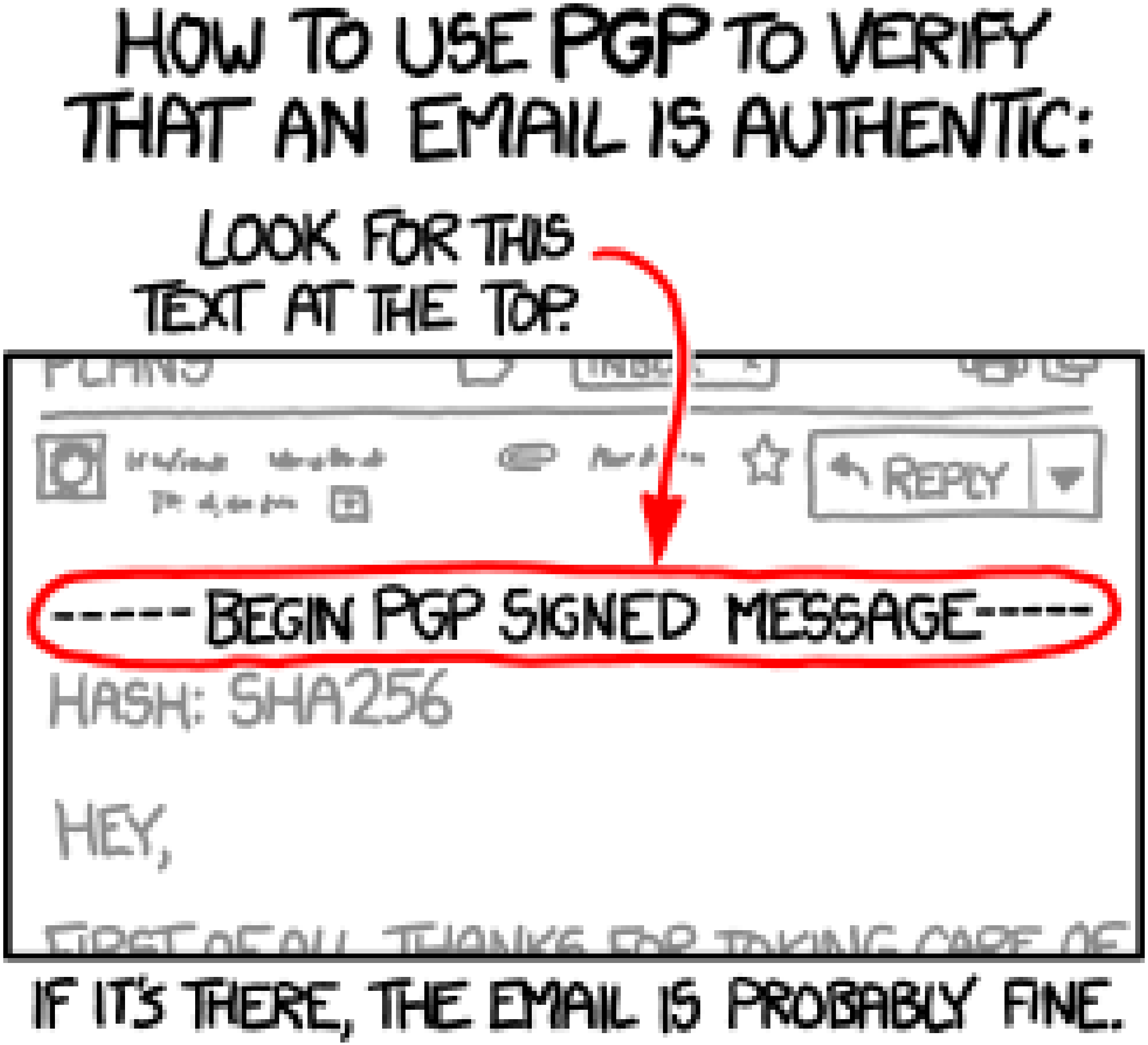
Provider	SPF Policy	DMARC Policy
Gmail	soft fail	none
Yahoo	neutral	reject
Outlook	soft fail	none
iCloud	soft fail	none
Hushmail	soft fail	–
Lycos	soft fail	–
Mail.com	fail	–
Zoho	soft fail	–
Mail.ru	soft fail	none
AOL	soft fail	reject
QQ	soft fail	none
Me.com	soft fail	none
Facebook	fail	reject
GoDaddy	fail	none
Yandex	soft fail	–
OVH	neutral	–
Comcast	neutral	none
AT&T	–	–
Verizon	neutral	–

Table 17: **SPF and DMARC Policies**—The majority of popular mail providers we tested posted an SPF record, but only three used the “strict fail” policy. Even fewer providers posted a DMARC policy, of which only three used “strict reject.”

PGP?

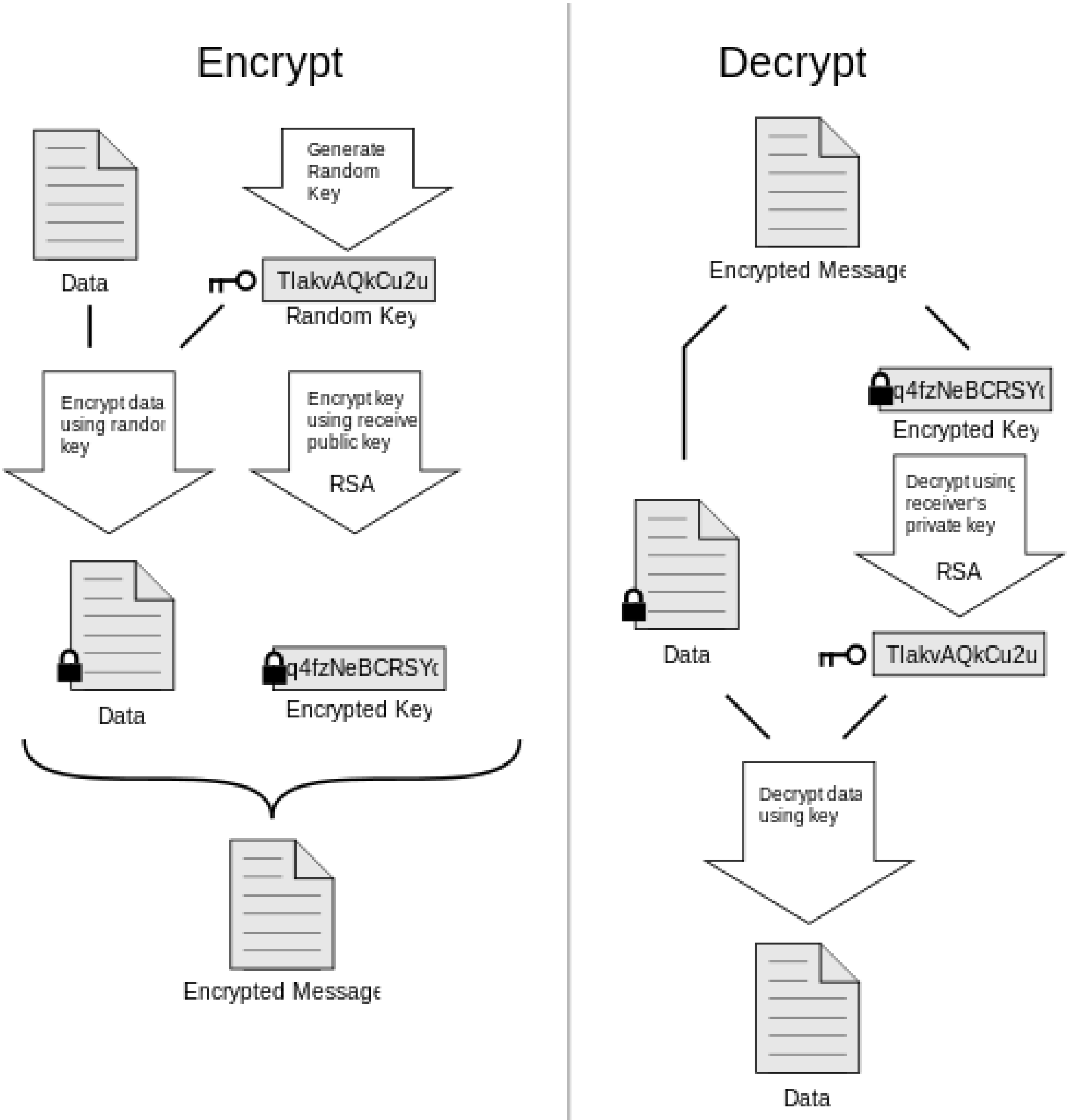


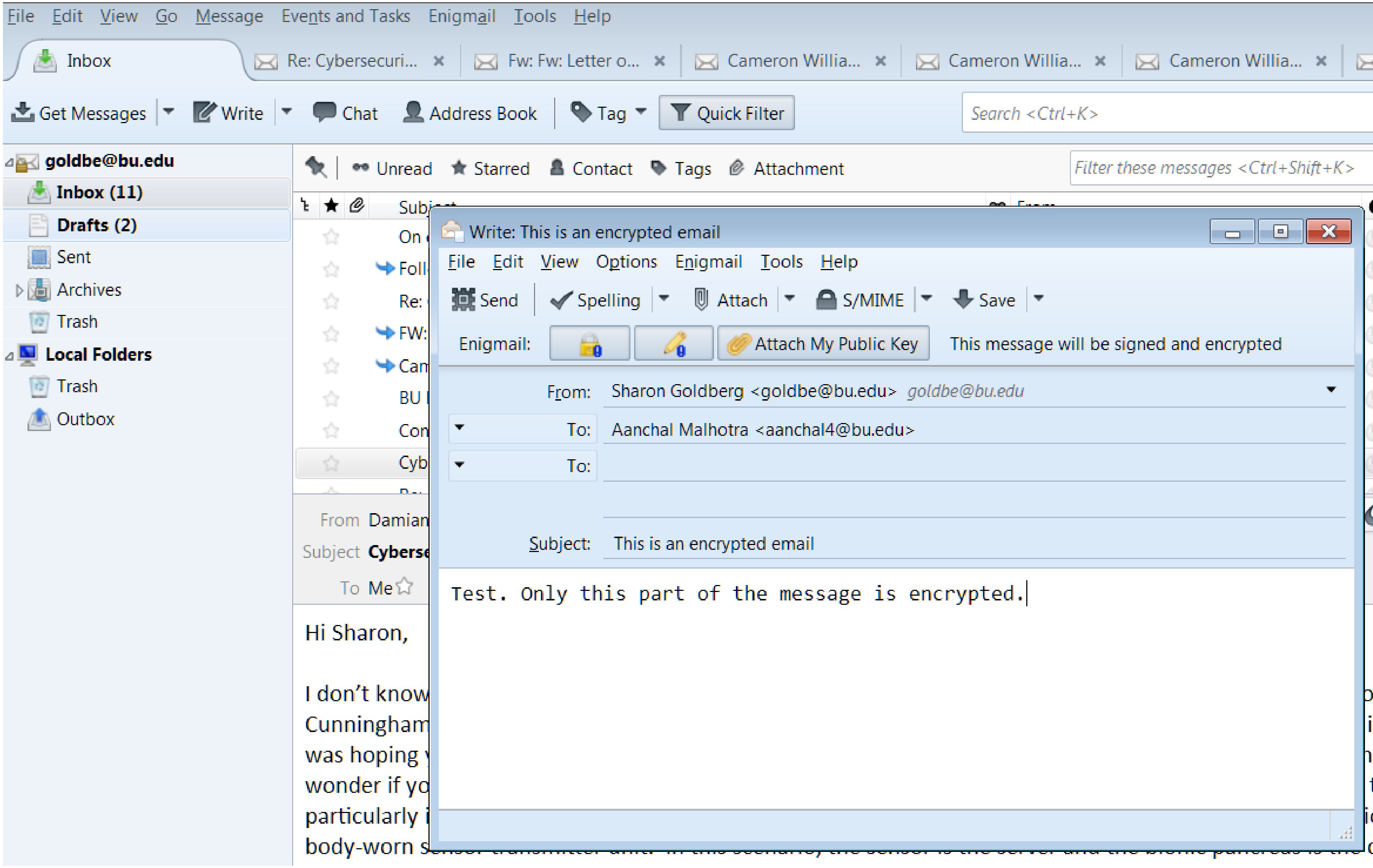
Finally, we note that end-to-end mail encryption, as provided by PGP [4] and S/MIME [42], does not address many of the challenges we discuss in this work. While these solutions do safeguard message content, they leave metadata, such as the subject, sender, and recipient, visible everywhere along the message's path. This information is potentially exposed to network-based attackers due to the lack of robust confidentiality protections for SMTP message transport. Although greater adoption of end-to-end encryption would undoubtedly be beneficial, for now, the overwhelming majority of messages depend solely on SMTP and its extensions for protection.


HOW TO USE PGP TO VERIFY THAT AN EMAIL IS AUTHENTIC:
LOOK FOR THIS TEXT AT THE TOP.
PLANS
INBOX
REPLY
-----BEGIN PGP SIGNED MESSAGE-----
HASH: SHA256
HEY,
FIRST OF ALL, THANKS FOR TAKING CARE OF
IF IT'S THERE, THE EMAIL IS PROBABLY FINE.

https://xkcd.com/1181/

PGP encryption of message contents





This message will be signed and encrypted

From: Sharon Goldberg <goldbe@bu.edu> *goldbe@bu.edu*

To: Aanchal Malhotra <aanchal4@bu.edu>

To:

Subject: This is an encrypted email

```
Test. Only this part of the message is encrypted.
```

From Damian

Subject Cyberse

To Me

Hi Sharon,

I don't know

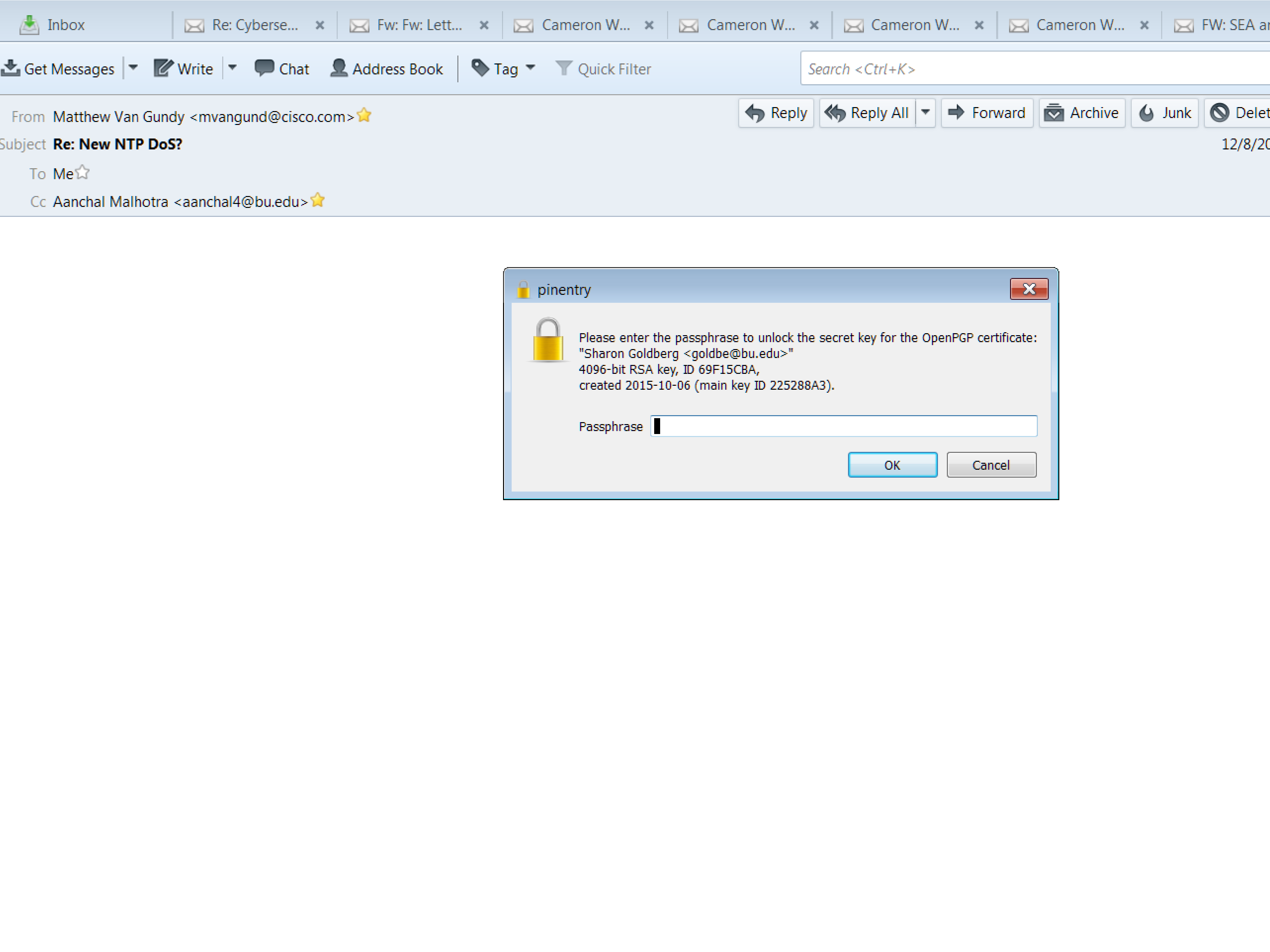
Cunningham

was hoping

wonder if yo

particularly

body-worn s



From Matthew Van Gundy <mvangund@cisco.com>

Subject **Re: New NTP DoS?**

12/8/20

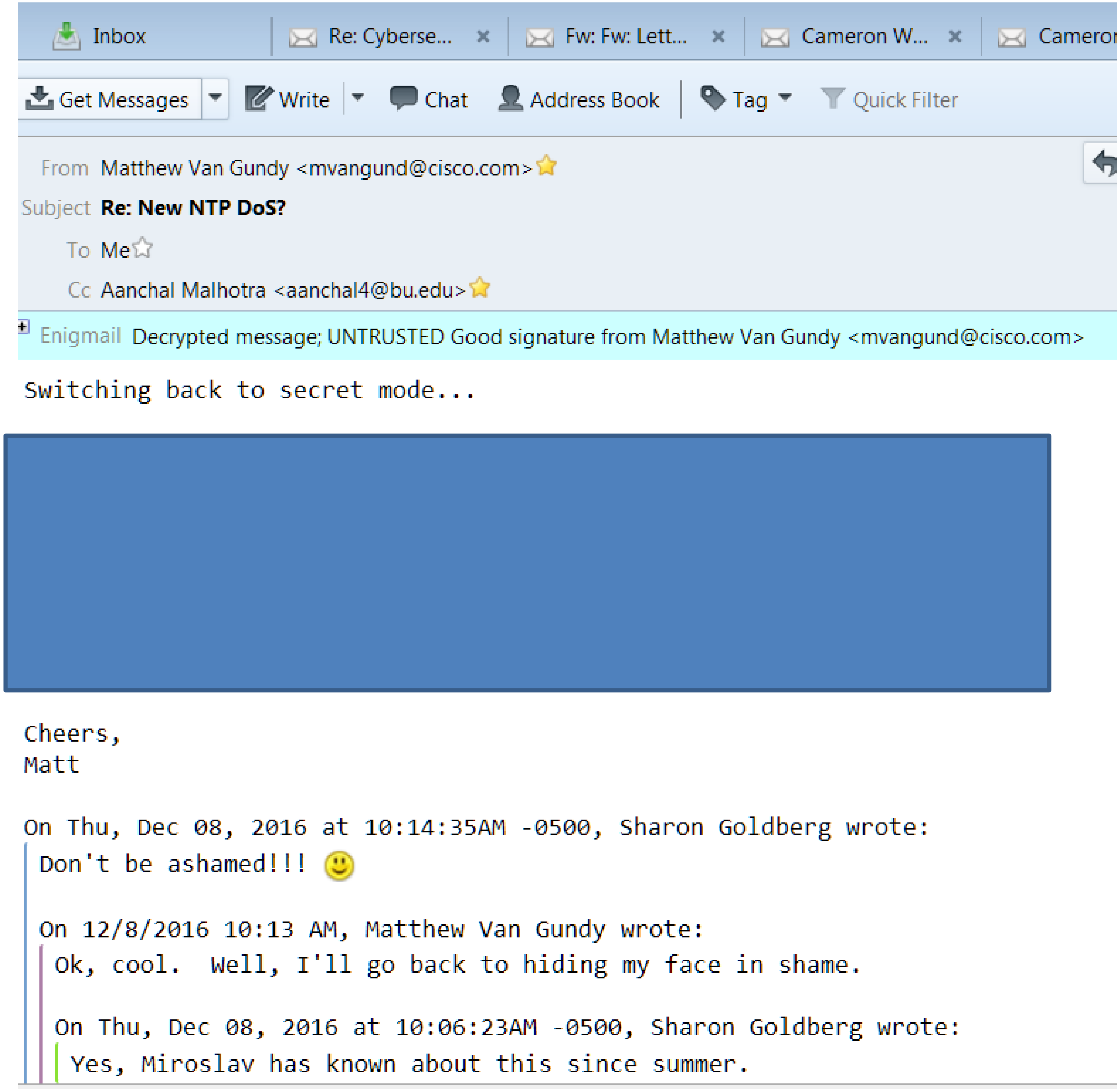
To Me

Cc Aanchal Malhotra <aanchal4@bu.edu>

pinentry

Please enter the passphrase to unlock the secret key for the OpenPGP certificate:
"Sharon Goldberg <goldbe@bu.edu>"
4096-bit RSA key, ID 69F15CBA,
created 2015-10-06 (main key ID 225288A3).

Passphrase



From Matthew Van Gundy <mvangund@cisco.com>

Subject **Re: New NTP DoS?**

To Me

Cc Aanchal Malhotra <aanchal4@bu.edu>

Enigmail Decrypted message; UNTRUSTED Good signature from Matthew Van Gundy <mvangund@cisco.com>

```
Switching back to secret mode...
```

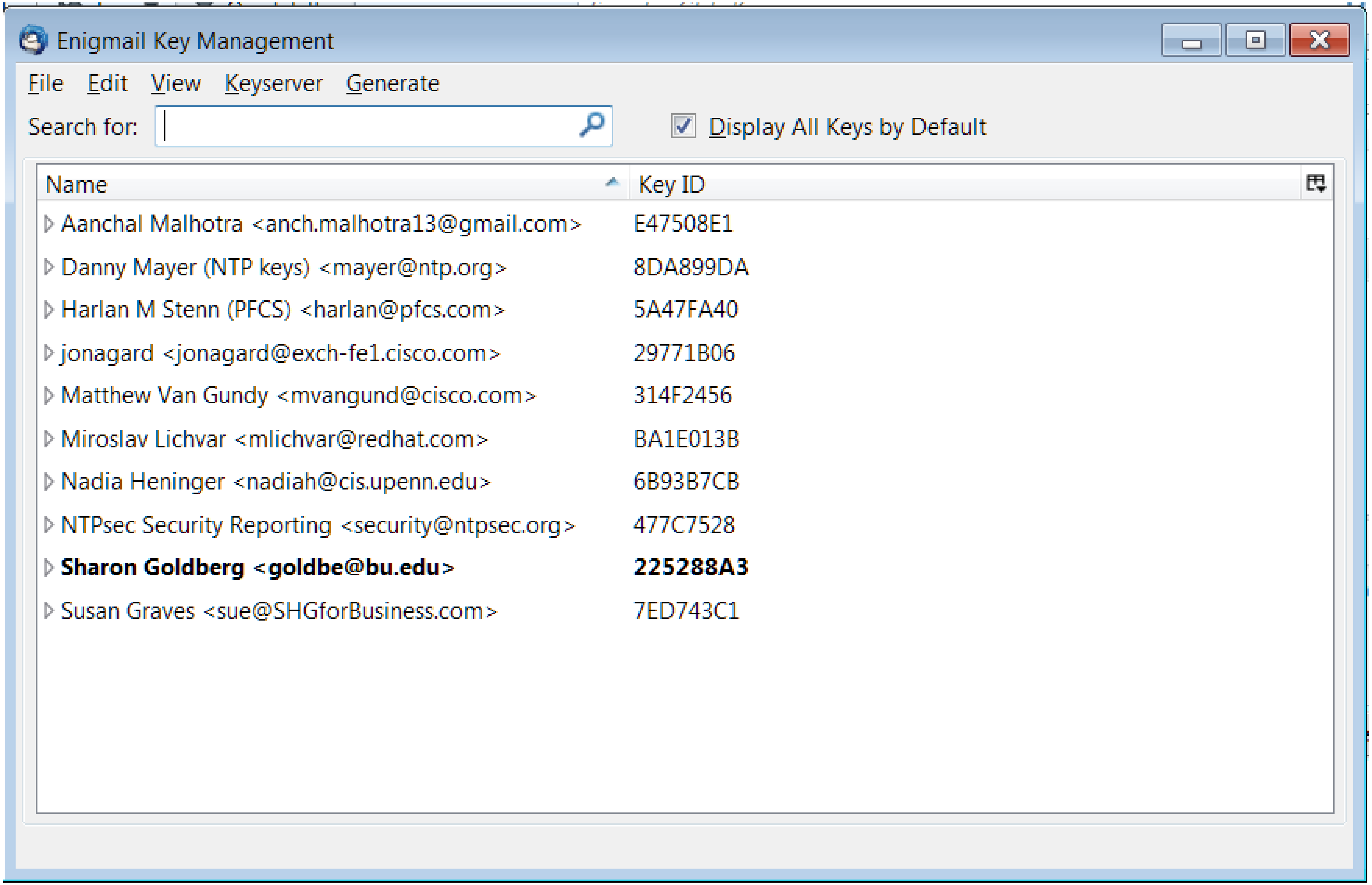
[redacted]

```
Cheers,
Matt

On Thu, Dec 08, 2016 at 10:14:35AM -0500, Sharon Goldberg wrote:
```

> Don't be ashamed!!! 🙂
>
> On 12/8/2016 10:13 AM, Matthew Van Gundy wrote:
>
> > Ok, cool. Well, I'll go back to hiding my face in shame.
> >
> > On Thu, Dec 08, 2016 at 10:06:23AM -0500, Sharon Goldberg wrote:
> >
> > > Yes, Miroslav has known about this since summer.

PGP keys

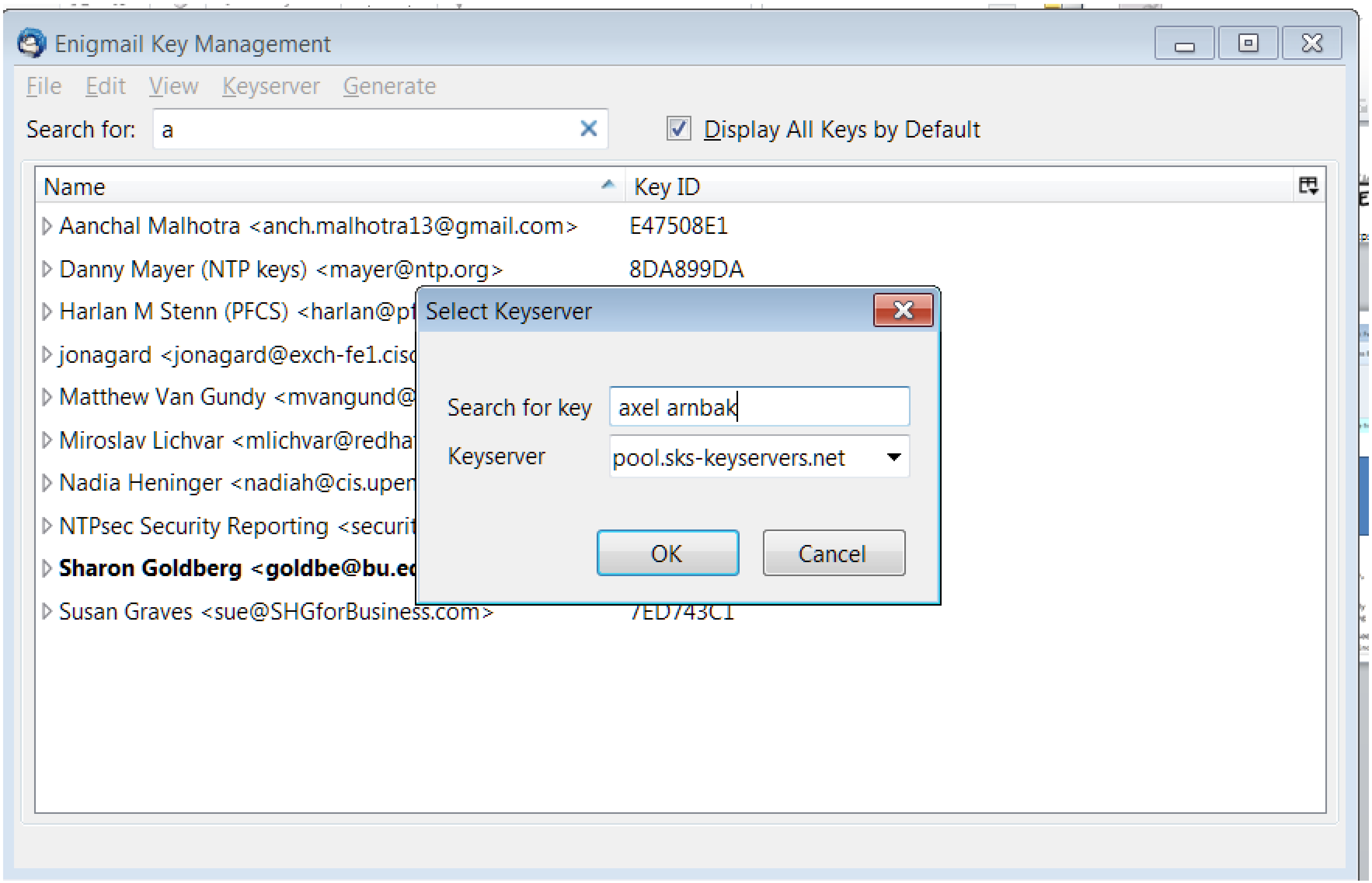


Search for:

☑ Display All Keys by Default

Name	Key ID
Aanchal Malhotra <anch.malhotra13@gmail.com>	E47508E1
Danny Mayer (NTP keys) <mayer@ntp.org>	8DA899DA
Harlan M Stenn (PFCS) <harlan@pfcs.com>	5A47FA40
jonagard <jonagard@exch-fe1.cisco.com>	29771B06
Matthew Van Gundy <mvangund@cisco.com>	314F2456
Miroslav Lichvar <mlichvar@redhat.com>	BA1E013B
Nadia Heninger <nadiah@cis.upenn.edu>	6B93B7CB
NTPsec Security Reporting <security@ntpsec.org>	477C7528
Sharon Goldberg <goldbe@bu.edu>	**225288A3**
Susan Graves <sue@SHGforBusiness.com>	7ED743C1

PGP keys



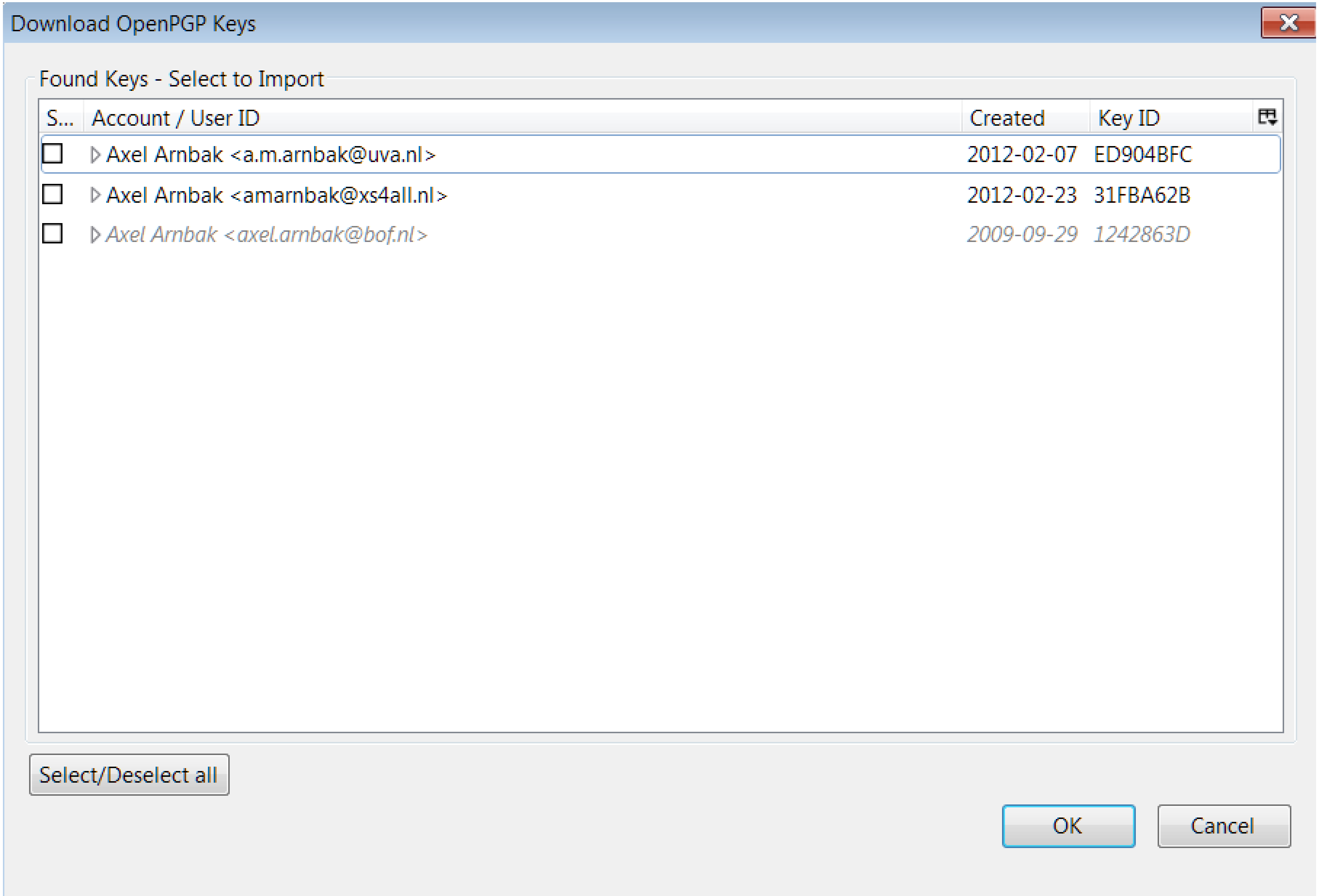
Search for: a

☑ Display All Keys by Default

Name	Key ID
Aanchal Malhotra <anch.malhotra13@gmail.com>	E47508E1
Danny Mayer (NTP keys) <mayer@ntp.org>	8DA899DA
Harlan M Stenn (PFCS) <harlan@pf	
jonagard <jonagard@exch-fe1.cisc	
Matthew Van Gundy <mvangund@	
Miroslav Lichvar <mlichvar@redha	
Nadia Heninger <nadiah@cis.upen	
NTPsec Security Reporting <securit	
Sharon Goldberg <goldbe@bu.ed	
Susan Graves <sue@SHGforBusiness.com>	7ED743C1

Search for key: axel arnbak

Keyserver: pool.sks-keyservers.net



S...	Account / User ID	Created	Key ID
□	▷ Axel Arnbak <a.m.arnbak@uva.nl>	2012-02-07	ED904BFC
□	▷ Axel Arnbak <amarnbak@xs4all.nl>	2012-02-23	31FBA62B
□	▷ *Axel Arnbak <axel.arnbak@bof.nl>*	*2009-09-29*	*1242863D*