

CVTR-2025-002

How to Manually Verify Email PGP Signatures Using GPG Tools

Zaid Khaishagi

Sr. Technical Associate, hello@crimsonvista.com

Follow this link to find similar works: <https://www.crimsonvista.com/crimson-labs>

How to Manually Verify Email PGP Signatures Using GPG Tools

This blog post will guide you through the process of manually verifying PGP signatures on emails using GPG Tools. Ensuring the authenticity and integrity of your digital communications is crucial in today's world. PGP (Pretty Good Privacy) provides encryption and signing capabilities that allow users to verify that an email truly came from the claimed sender and has not been tampered with during transit.

I'll explain how you can generate email signatures (and encryption) yourself and how you can verify a signed email. So feel free to skip to the part relevant to you.

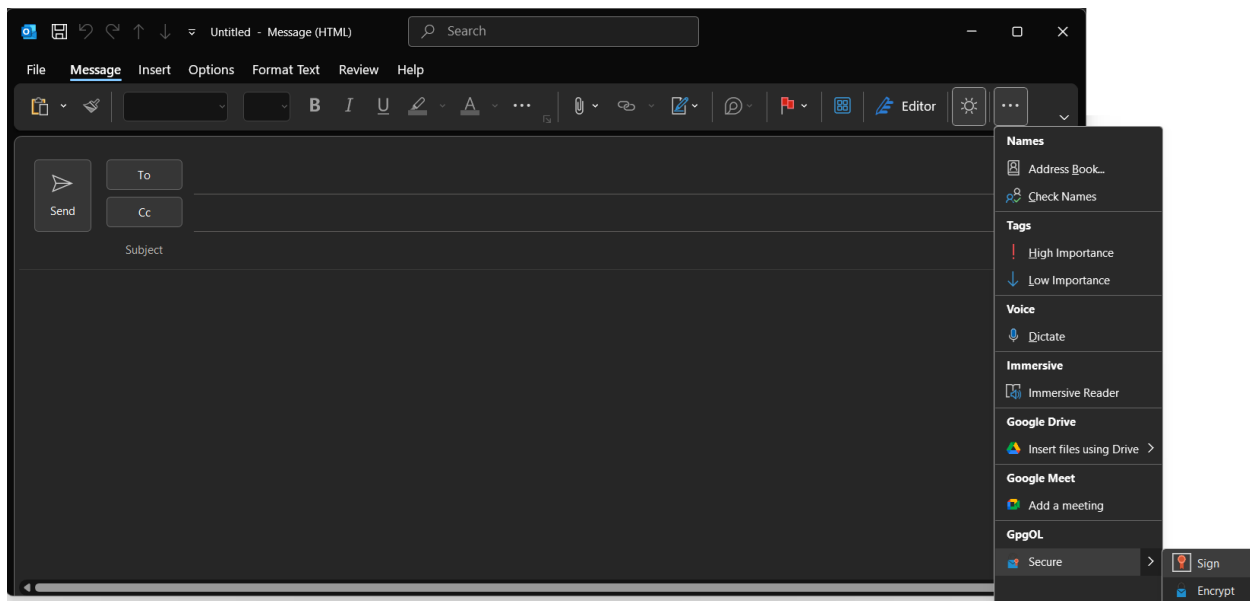
How to sign emails

Users can use the GpgOL extension in order to generate signatures when they send an email. Once you have Outlook installed on your device, you can install the GpgOL extension by going to <https://www.gpg4win.org/index.html> and downloading the installer from there. It will ask you about which components you would like to install. You should definitely install Kleopatra because that is the key manager which will manage your keys for signing and encrypting emails. This is important because these keys are what let you sign/encrypt emails as you send them and verify/decrypt emails as you receive them. You should also install GpgOL which is the extension for Outlook. You may choose any of the other options as well if you like them.

Once you have installed these components, you may need to restart your computer in order to see the Outlook extension appear.

After installing the Outlook extension, you can see the GpgOL options to either sign or encrypt a message when you compose a new email by going to the horizontal ellipses icon and expanding the 'Secure' options under GpgOL and selecting either 'Sign' or 'Encrypt'. You can also select both of them (the white outline on the icon indicates whether it is selected).

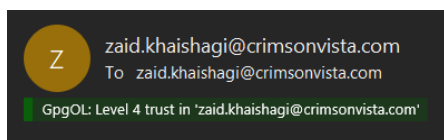
Now, when you send the email, it will apply the signature or encryption according to the options you selected.



How to verify emails manually

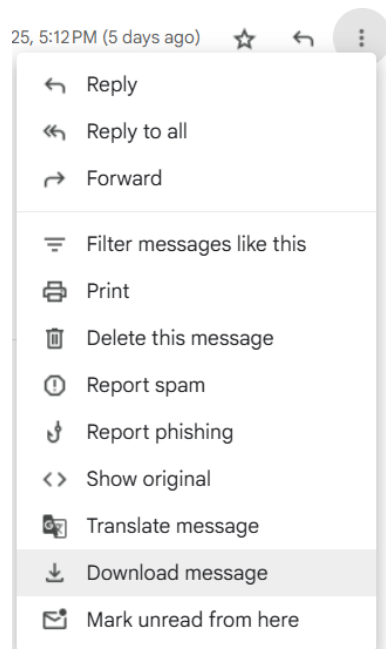
When you receive a signed email, you can do it either by using the GpgOL extension which will automatically verify it for you or by doing it manually using the command line gpg tools (installed with gpg4win).

If you have the public key of the sender saved in Kleopatra, Outlook will automatically verify the email signature. To test this and for demonstration purposes, I sent a signed email to myself. The verification shows up like this:



In order to manually verify the email, you will need the public key of the sender saved into Kleopatra (the key manager). You can add it to Kleopatra through the GUI or also through the command line. Once you have done this, you need to first download the signature, which is an .asc file. In this example, the filename is "openpgp-digital-signature.asc". If you are using Gmail, this will show up as an attachment.

You also need to download/save the raw email message as a '.eml' file. If you are using Gmail, you can do this by clicking on the "kebab" menu (the 3 vertical dots) and then clicking on 'Download Message'.



After downloading these 2 files, you can open them with any text editor to see their contents.

Open the saved email .eml file in a text editor and look for the first occurrence of the email header 'boundary'. It will look like this: `boundary="==I47ptrHg5tFQrZ=="`. This boundary string is used in the email to separate the different parts of the email. When it is used to separate email parts, it is preceded by two dashes, so it looks like this: `--==I47ptrHg5tFQrZ==`.

The first part of the email, from the start of the email file to the first use of the boundary string contains the email headers. The second part, between the first and second use of the boundary string, is the email body. In a signed email (without any more attachments) like in this example, the part between the second and the third use of the boundary string contains the signature.

In a multipart email, there may be additional boundary strings mentioned inside the email body (which was the case in this example). In this situation, it may look something like this:

```
□--==I47ptrHg5tFQrZ==
Content-Type: multipart/alternative;
    boundary=="==BaRW6lPAQpve80=="
```

□

Here, the first (top level) boundary string marks the start of the email body. Then the other boundary string is mentioned which is used within the email body to separate different parts within the email body. It will not be used outside the email body.

Now, you need to copy the email body exactly as it was into a new text file and save it. You should do this by starting at the first use of the top-level boundary string and ending at the second use of the top-level boundary string (the boundary for the email body). When you copy,

do not copy the top-level boundary string itself; start at the next line (should be a blank line) from the boundary string and end at the line just above the boundary string (should be a blank line). It is crucial that you copy the correct portion without leaving out anything, because it will not verify otherwise.

Once you have copied it into a new text file, save it.

Now, you can verify the email manually by using this command:

```
❏ gpg --verify <signature_file>.asc <email_body>.txt
```

- ❏ <signature_file>.asc : This is the signature file saved as a .asc file
- ❏ <email_body>.txt : This is the email body part copied from the saved email .eml file and saved in a new text file.

Note: If you see any errors, try to open the signature file and remove any trailing new lines from the end and try again.

When you run this command, you should see a message saying that the email was successfully verified with a good signature. For example, it should look like this:

```
C:\Users\Zaid\OneDrive\Desktop>gpg --verify openpgp-digital-signature.asc test.txt
gpg: Signature made 05/21/25 17:36:51 Pacific Daylight Time
gpg:                using EDDSA key CED267[REDACTED]667
gpg: Good signature from "Zaid Khaishagi <zaid.khaishagi@crimsonvista.com>" [ultimate]
```