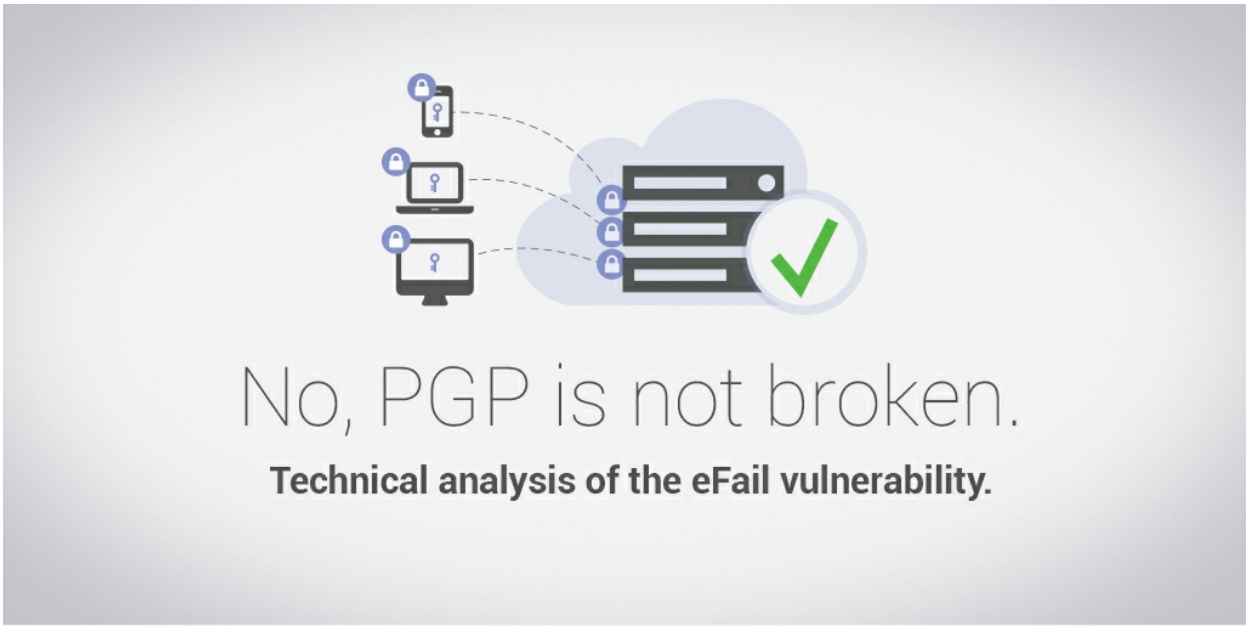
# No, PGP is not broken, not even with the Efail vulnerabilities

protonmail.com/blog/pgp-vulnerability-efail/

Andy Yen May 15, 2018




## Recently, news broke about potential vulnerabilities in PGP, dubbed Efail. However, despite reports to the contrary, PGP is not actually broken, as we will explain in this post.

The vulnerability report, which came with its own website, efail.de, has attracted a lot of headlines such as the one below, along with recommendations to disable the usage of PGP plugins. This was not helped by the fact that until earlier today, the full details of the "vulnerability" were not disclosed to the community. However, now that the information is public and we have completed our analysis, we can say that **these headlines are wrong**, and recommendations to stop using PGP plugins are misguided. We have published our own recommendations below as well.

**First, ProtonMail is not impacted by the Efail PGP vulnerabilities.** This includes our web and mobile applications, and also our ProtonMail Bridge software for desktop. In the second part of this post, we have included a full technical analysis explaining why ProtonMail is not vulnerable.

**It is equally important to state that, other than one minor exception (discussed later), Efail is not a vulnerability in PGP itself.** What the authors of Efail did was catalogue a list of PGP clients that have errors in their PGP implementation. In the worst case, these implementation flaws can lead to encrypted emails being decrypted, provided the attackers already have a copy of your encrypted email and the capability to resend it to you (both are pretty strong caveats that dramatically limit the scope of the vulnerability).

### PGP Security and Efail

PGP has a long history, dating back over 20 years, and while some may use this to claim that PGP is "outdated" or "unfashionable", it also means that PGP is time and battle-tested. Simply put, it is the best way to encrypt emails, and because the protocol is open, it has had over two decades of peer review and security audits. This means it is very unlikely to have undiscovered vulnerabilities lurking under the hood.

However, in cryptography, as in many things, the devil is in the details, and implementing PGP securely is indeed a challenge. Many of the PGP software libraries and plugins have not undergone the same level of scrutiny as the OpenPGP protocol itself. Indeed, some of the vulnerabilities disclosed in Efail have been known to the PGP developer community since 1999 and some PGP plugins remain vulnerable.

As an open standard, anybody can implement PGP, and some do it better than others, so it should come as no surprise that some PGP implementations have security vulnerabilities. However, this should not be taken to be an indictment of PGP. **At its core, PGP remains cryptographically sound**, and using a few bad implementations to claim that "PGP has a serious flaw" is both untrue and disingenuous.

## Recommendations for PGP users

**Recommendations to disable PGP plugins and stop encrypting emails are completely unwarranted and could put lives at risk.** The correct response to vulnerable PGP implementations should not be to stop using PGP, but to use secure PGP implementations. If a vulnerability is discovered in your operating system, you don't throw away your computer. Instead, you update it and patch it. When it comes to vulnerabilities in PGP implementations, the same principle applies. If you are a PGP user, we recommend the same strategy. **Apply updates to your PGP software when they become available** (if necessary). Because the vulnerabilities are in the PGP implementations and not the OpenPGP protocol itself, these bugs are very easy for PGP plugin developers to patch. **Or you can switch to using ProtonMail which is not susceptible to the Efail vulnerabilities.**

## ProtonMail is not impacted by the Efail PGP vulnerabilities

In this section, we will provide a technical analysis of why ProtonMail is not impacted by the PGP implementation vulnerabilities disclosed in the Efail paper. Due to the subject matter, the discussion is necessarily technical in nature. At the end, we also discuss our views on the future of PGP.

There are three distinct attacks presented in the paper – a direct exfiltration attack, an attack on S/MIME, and an attack on OpenPGP. We have analyzed the first and third for any potential vulnerabilities, as ProtonMail does not use or support S/MIME. We will note, however, that S/MIME is actually the more serious vulnerability because it is widely used by government and military and may be unfixable, so the media's fixation on PGP is misplaced since PGP itself is not actually broken.

**Direct Exfiltration Channel Attack**

The first attack presented in the paper is the “direct exfiltration channel” attack, in which an adversary constructs an email structure where a plaintext part with an <img> element is prepended to the ciphertext. Then, when the ciphertext is decrypted by the recipient, it will theoretically be replaced by the corresponding plaintext in the client, and then the client will attempt to parse the image source, sending the plaintext in the HTTP request path.

ProtonMail is not vulnerable to this attack because we do not handle emails containing both plaintext and ciphertext in the way described in the paper. When ProtonMail’s servers parse incoming mail, a message of this structure is treated as a plaintext email and encrypted with the recipient’s public keys before being sent to ProtonMail clients. The only case in which the message is not encrypted by the server before being sent to the client is if the message begins with a valid ASCII-armored PGP message. In this case, the PGP message is used directly and any subsequent plaintext is discarded. This ensures that all emails sent to the clients are encrypted and cannot be a mixture of plaintext and ciphertext.

When a message is opened, the ProtonMail client attempts to decrypt the entire PGP message from the server using the recipient’s private keys. If decryption succeeds and the decrypted content contains any additional encrypted content, the client will never attempt to decrypt this “nested” content. Therefore, an attempted attack of this type would merely result in the encrypted text being sent in the HTTP request path (which, according to the attack parameters, the attacker would have already), and only if the user chose to load remote content, which is always disabled by default. Additionally, because ProtonMail’s clients never do any “nested” decryption, it would be impossible for ProtonMail’s servers to launch this type of attack.

**Malleability Gadget Exfiltration Channel Attack**

The “malleability gadget exfiltration channel” attack relies on the fact that due to the malleability of Cipher Feedback Mode (CFB), the mode of encryption used by OpenPGP, knowledge of some of a ciphertext’s decrypted content can allow an attacker to inject malicious content into the ciphertext itself.

This attack is circumvented by the use of the Symmetrically Encrypted with Integrity Protection (SEIP) OpenPGP packet for storing encrypted data, which ProtonMail has always used for encrypted emails sent between ProtonMail users, and which is used by the vast majority of modern PGP clients. When data is encrypted with this packet type, the plaintext data that is about to be encrypted is hashed and stored in a Modification Detection Code (MDC) packet. During the decryption process, the decrypted data is hashed and compared with the result stored in the MDC packet. If the results are not equal, this is treated by all of ProtonMail’s OpenPGP libraries as a decryption failure, and clients will not load the decrypted content.

The paper brings up three methods by which integrity protection could be defeated. The first and second, Ignoring the MDC and Stripping the MDC, are not feasible in ProtonMail’s clients because our cryptography libraries would treat these scenarios as decryption failures. The third method is for the attacker to change an SEIP packet into a Symmetrically Encrypted packet, which is an obsolete OpenPGP packet with no modification detection capabilities.

This is the minor exception that we mentioned previously. While this can be construed as a vulnerability in the OpenPGP standard, the solution is easy (drop support for this obsolete packet type). The open-source library used by our web application, OpenPGP.js, is not vulnerable to this attack because it already does not allow the use of Symmetrically Encrypted packets with any AES ciphers, which are the only ciphers supported for symmetric encryption by ProtonMail and most PGP clients. Even without this, ProtonMail users are protected from this attack because ProtonMail servers disallow all OpenPGP packets of the Symmetrically Encrypted type from all mail.

## The Future of PGP

**PGP remains to this day, the best way for encrypting emails, and if implemented properly, is both secure and reliable.** While ProtonMail was not impacted by the recent PGP implementation vulnerabilities, we recognize that as the leaders in this space, we also need to play a part in ensuring that other PGP implementations are also secure. In pursuit of this goal, and as part of our responsibilities as the maintainers of OpenPGPjs, we are doing the following:

First, we are dropping support of the obsolete Symmetrically Encrypted packet type when OpenPGPjs is used with default options. As OpenPGPjs is one of the world's most widely used OpenPGP libraries (powering popular plugins such as Enigmail, Mailvelope, and many others), this will help to protect the PGP ecosystem from Malleability Gadget Exfiltration Channel Attacks in the future.

Secondly, we will continue to push forward Authenticated Encryption. Authenticated encryption with associated data (AEAD) is a form of encryption that provides confidentiality, integrity, and authenticity guarantees on data, evading any attack of the types described in the paper. The Efail paper erroneously claims that "there were efforts to introduce authenticated encryption in OpenPGP which are, however, expired." This is untrue, as there is currently a working draft for AEAD support in OpenPGP, which has been gaining significant traction in the security community. We have also recently completed a full implementation of AEAD in OpenPGP.js which is currently undergoing a third party security audit.

As the world's largest secure email provider, we take security very seriously and will continue to innovate to ensure that your data remains secure and private, both today and in the future.

Best Regards,
The ProtonMail Team

For the official statement from the maintainers of GNUPG, another common PGP software, please see here.

*The above analysis includes contributions from Sanjana Rajan, Bart Butler, Andy Yen, Daniel Huigens and Yanfeng Zhang from the ProtonMail team.*

***ProtonMail provides free encrypted email accounts to the public.***

***We also provide a free VPN service to protect your privacy.***

Share This!

Get a Free Encrypted Email Account