TRAIL OF BITS

die, PGP, die 🔪

William Woodruff
Trail of Bits

hello

i am:

- **william woodruff (@8x5clPW2)**
- **senior security engineer @ Trail of Bits**
  - LLVM, applied cryptography, open source engineering

agenda:

- **PGP is bad**
  - real bad
- **“i can fix him”**
  - no you can’t
- **what to do and to use instead**
  - tl;dr signal & age & sigstore (soon!)

your 5 second refresher on PGP

"Pretty" ""Good"" """Privacy"""

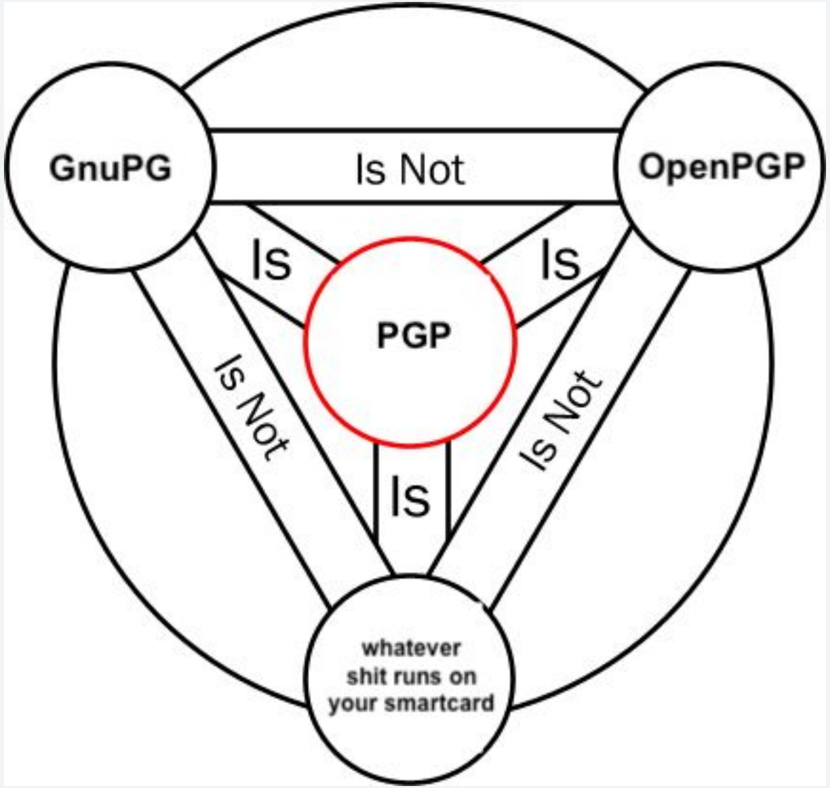
good for:

- **email encryption (not really)**
- **authenticated channels (not really)**
- **digital signing (not really)**

key features:

- **"web of trust"**
- **historical museum of bad crypto**

"standardized" in RFC 4880



let’s talk about cryptography

PGP is from 1991

George H. W. Bush vomiting incident

RFC 4880: your new best friend

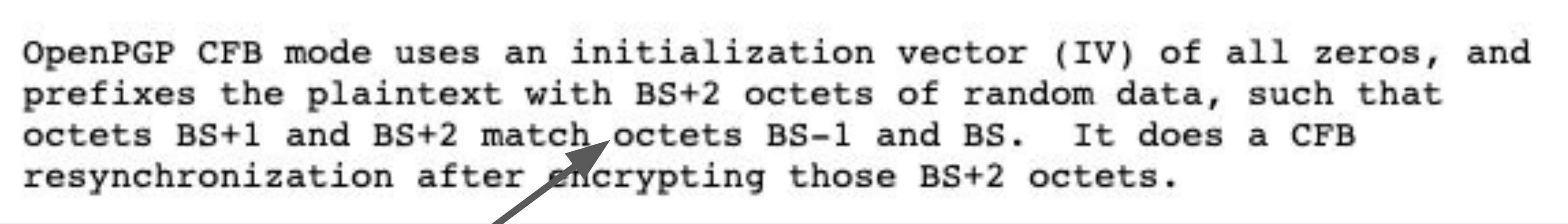
the original PGP RFC, describing the message format and baseline algorithms

- **public key: RSA, ElGamal, DSA (lol)**
- **symmetric:**
  - IDEA (weird obsolete 90s replacement for DES)
  - 3DES (your bank loves this one)
  - CAST5 (the official block cipher of 🇨🇦)
  - Blowfish (broken)
  - Twofish (okay? maybe? who knows?)
  - AES (good, but **marked as optional**)
- **extended by 5581 (Camellia) and 6637 (ECC)**
  - better(?), but both optional! the baseline is still 4880!

who needs GCM?

want a sane mode of operation? too bad, you only get PGP’s weird custom CFB

psst: it’s not authenticated

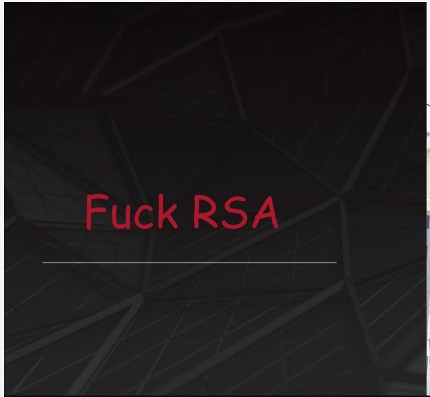


OpenPGP CFB mode uses an initialization vector (IV) of all zeros, and prefixes the plaintext with BS+2 octets of random data, such that octets BS+1 and BS+2 match octets BS-1 and BS. It does a CFB resynchronization after encrypting those BS+2 octets.

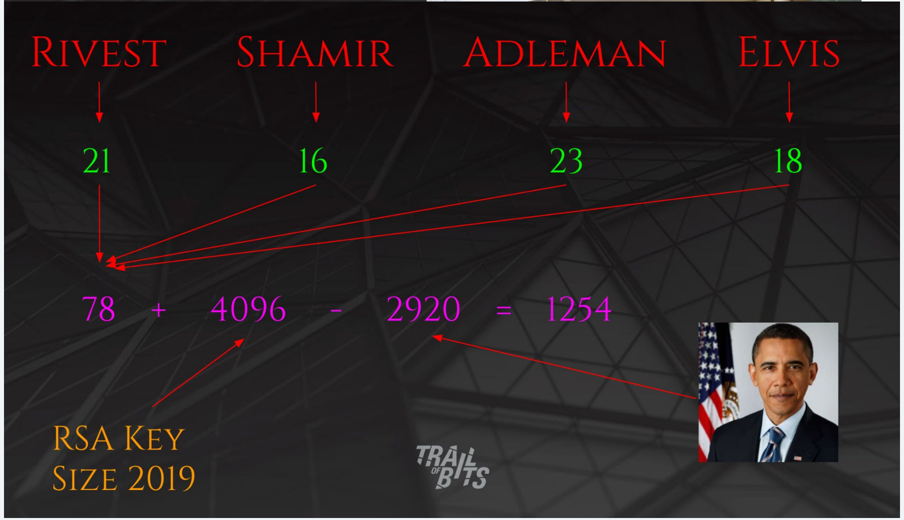
statements dreamed up by the utterly deranged

RSA (throwback!)

RSA



Fuck RSA

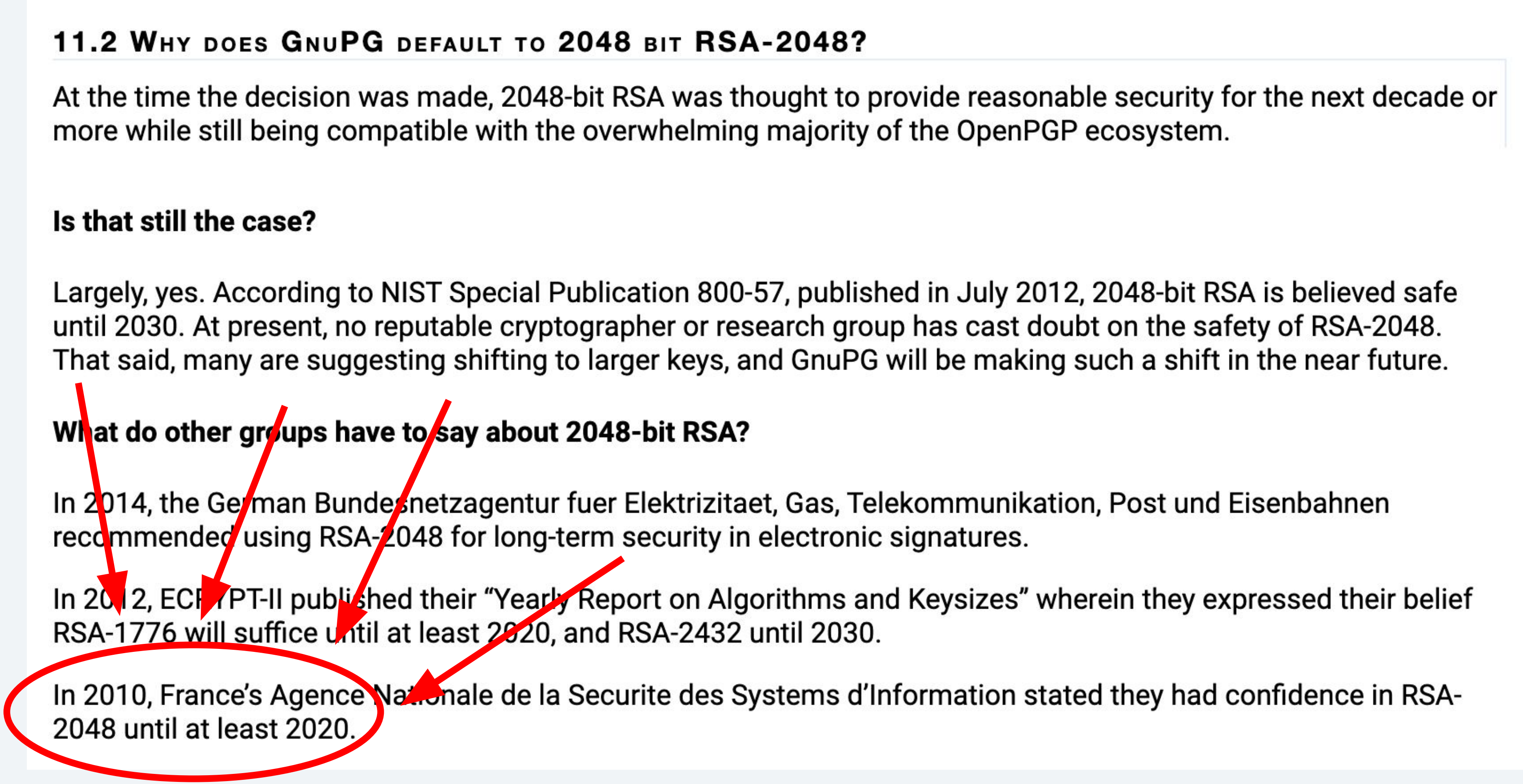


RIVEST SHAMIR ADLEMAN ELVIS

21 16 23 18

78 + 4096 - 2920 = 1254

RSA KEY SIZE 2019



11.2 Why does GnuPG default to 2048 bit RSA-2048?

At the time the decision was made, 2048-bit RSA was thought to provide reasonable security for the next decade or more while still being compatible with the overwhelming majority of the OpenPGP ecosystem.

Is that still the case?

Largely, yes. According to NIST Special Publication 800-57, published in July 2012, 2048-bit RSA is believed safe until 2030. At present, no reputable cryptographer or research group has cast doubt on the safety of RSA-2048. That said, many are suggesting shifting to larger keys, and GnuPG will be making such a shift in the near future.

What do other groups have to say about 2048-bit RSA?

In 2014, the German Bundesnetzagentur fuer Elektrizitaet, Gas, Telekommunikation, Post und Eisenbahnen recommended using RSA-2048 for long-term security in electronic signatures.

In 2012, ECRYPT-II published their “Yearly Report on Algorithms and Keysizes” wherein they expressed their belief RSA-1776 will suffice until at least 2020, and RSA-2432 until 2030.

In 2010, France’s Agence Nationale de la Securite des Systems d’Information stated they had confidence in RSA-2048 until at least 2020.

ElGamal is *weird*

circa 1985 (the before times)

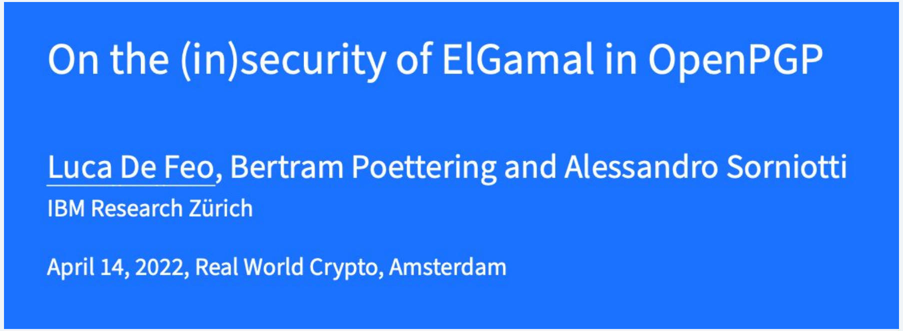
selected because of minimal legal encumbrance, not ideal cryptographic properties (seeing a pattern?)

in particular:

- malleability (CCA)
- you need to understand groups to achieve semantic security (like RSA)
- big ol’ keys for smol bit security (like RSA)
- essentially RSA but with discrete log instead of prime factorization

ElGamal is *weird*

did i mention that there's no actual standard for ElGamal?

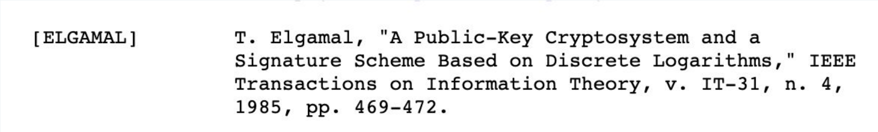


On the (in)security of ElGamal in OpenPGP

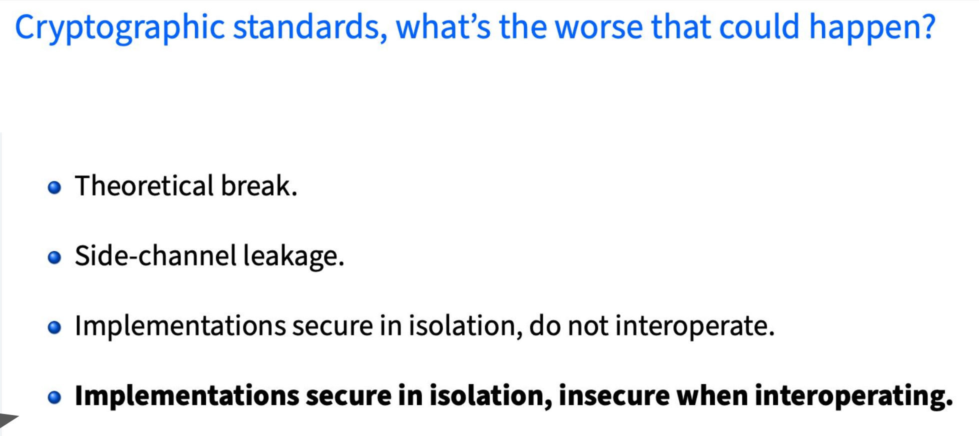
Luca De Feo, Bertram Poettering and Alessandro Sorniotti
IBM Research Zürich

April 14, 2022, Real World Crypto, Amsterdam

RFC4880 cites Taher's original paper:



```
[ELGAMAL]       T. Elgamal, "A Public-Key Cryptosystem and a
                Signature Scheme Based on Discrete Logarithms," IEEE
                Transactions on Information Theory, v. IT-31, n. 4,
                1985, pp. 469-472.
```

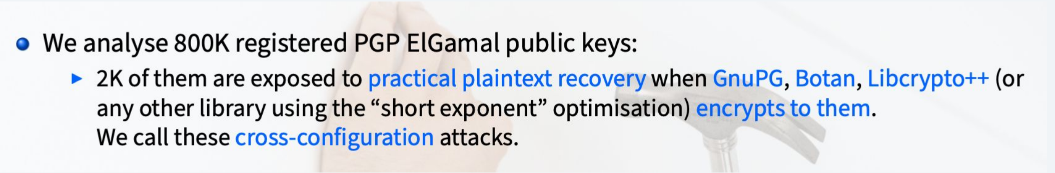


Cryptographic standards, what's the worse that could happen?

- Theoretical break.
- Side-channel leakage.
- Implementations secure in isolation, do not interoperate.
- **Implementations secure in isolation, insecure when interoperating.**

PGP implementations implement prime generation in different ways!

you are here



- We analyse 800K registered PGP ElGamal public keys:
  - 2K of them are exposed to practical plaintext recovery when GnuPG, Botan, Libcrypto++ (or any other library using the "short exponent" optimisation) encrypts to them.
    We call these cross-configuration attacks.

forward secrecy

“compromises in long term keys do not compromise short-lived sessions”

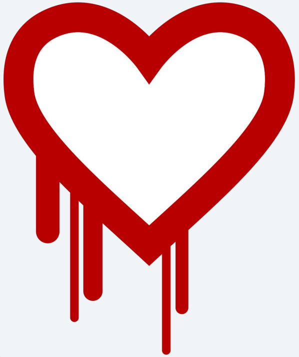
key compromise should not allow an adversary to decrypt passively captured historical traffic!

users ***expect*** this when communicating in a post-Heartbleed world!!

PGP cannot provide forward secrecy. there are no session keys.

key compromise in PGP means total compromise of all messages.

this is a completely solved problem in modern protocols (TLS 1.3, SSH, Noise)!!!



authenticated encryption

this is so fucking broken i’m not going to fill this slide in

google “PGP MDC”

“web of trust”

remember *key signing parties*?

“the strong set”

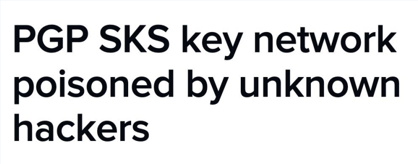
the set of keys such that each pair of keys has a path between them

in theory:

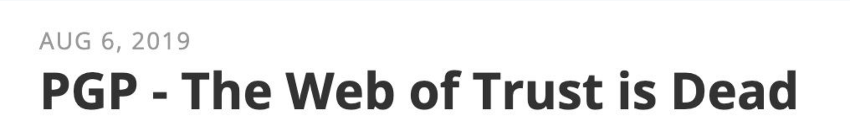
- **strong set continuously grows over time, strengthening the WoT**
- **more interconnections = more trust, amirite??**

in practice:

- **shaky/nonexist mechanisms for revocation/compromise**

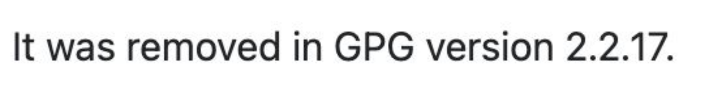


PGP SKS key network poisoned by unknown hackers



AUG 6, 2019

PGP - The Web of Trust is Dead

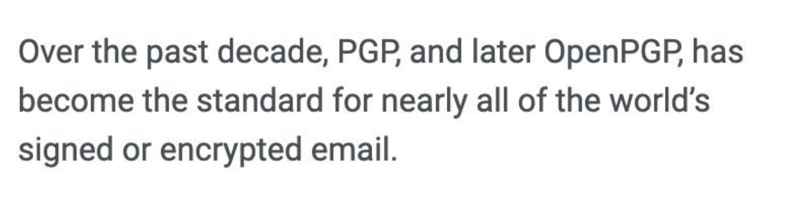


It was removed in GPG version 2.2.17.

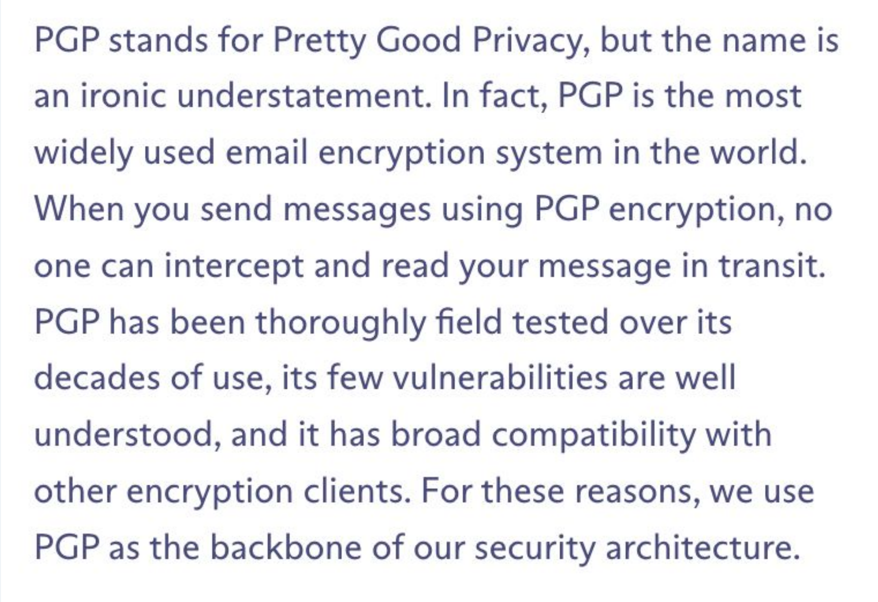
“Poisoned certificates are already on the SKS keyserver network. There is no reason to believe the attacker will stop at just poisoning two certificates. Further, given the ease of the attack and the highly publicized success of the attack, it is prudent to believe other certificates will soon be poisoned.”

let’s talk about email and PGP

how it started

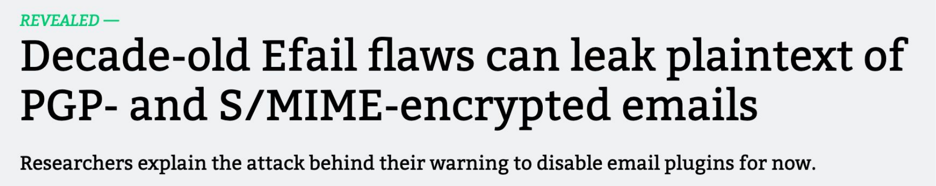


Over the past decade, PGP, and later OpenPGP, has become the standard for nearly all of the world's signed or encrypted email.



PGP stands for Pretty Good Privacy, but the name is an ironic understatement. In fact, PGP is the most widely used email encryption system in the world. When you send messages using PGP encryption, no one can intercept and read your message in transit. PGP has been thoroughly field tested over its decades of use, its few vulnerabilities are well understood, and it has broad compatibility with other encryption clients. For these reasons, we use PGP as the backbone of our security architecture.

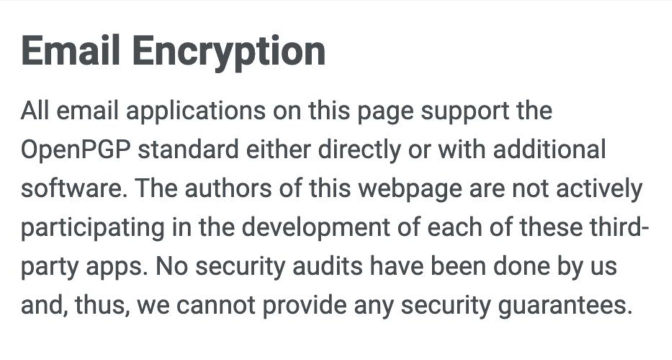
how it's going



REVEALED —

Decade-old Efail flaws can leak plaintext of PGP- and S/MIME-encrypted emails

Researchers explain the attack behind their warning to disable email plugins for now.

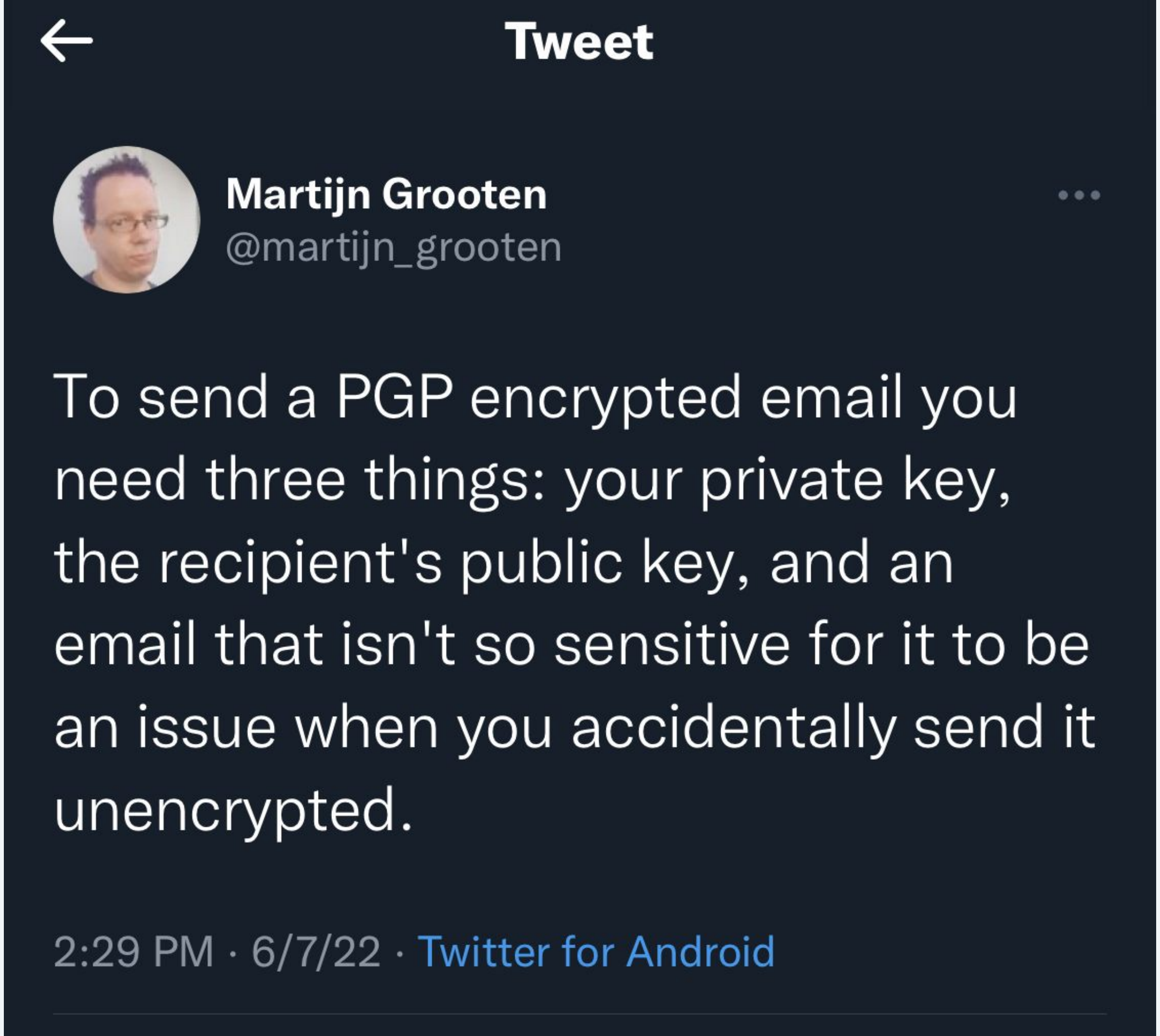


Email Encryption

All email applications on this page support the OpenPGP standard either directly or with additional software. The authors of this webpage are not actively participating in the development of each of these third-party apps. No security audits have been done by us and, thus, we cannot provide any security guarantees.

we cannot provide any security guarantees.

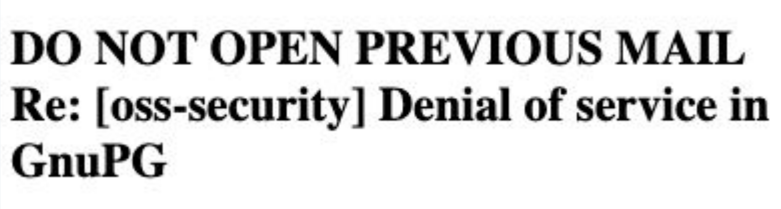
real heads know



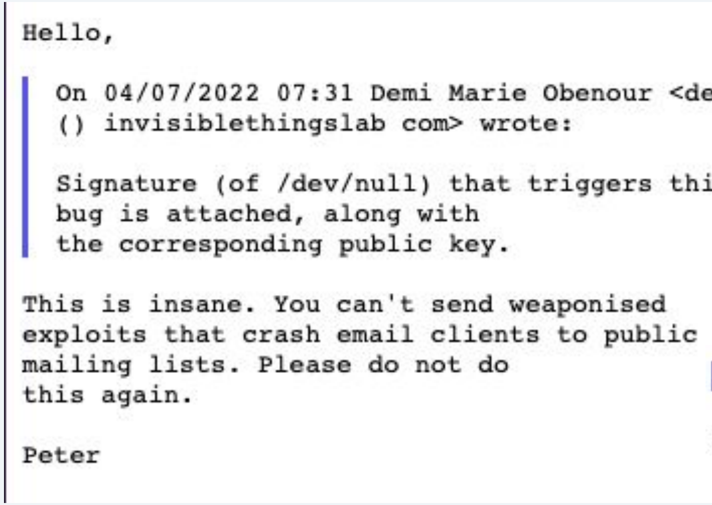
Martijn Grooten
@martijn_grooten

To send a PGP encrypted email you need three things: your private key, the recipient's public key, and an email that isn't so sensitive for it to be an issue when you accidentally send it unencrypted.

2:29 PM · 6/7/22 · Twitter for Android



DO NOT OPEN PREVIOUS MAIL
Re: [oss-security] Denial of service in GnuPG

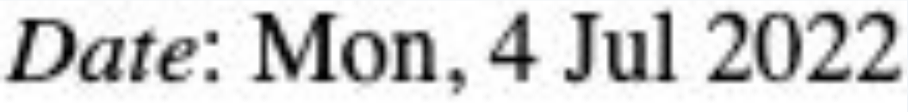


Hello,

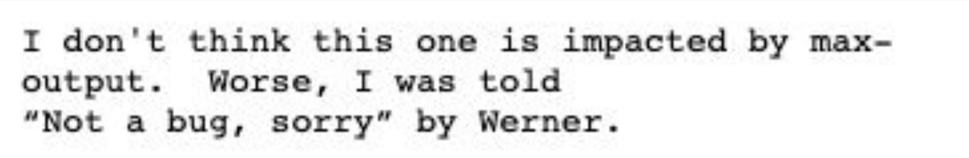
> On 04/07/2022 07:31 Demi Marie Obenour <demi () invisiblethingslab com> wrote:
>
> Signature (of /dev/null) that triggers this bug is attached, along with the corresponding public key.

This is insane. You can't send weaponised exploits that crash email clients to public mailing lists. Please do not do this again.

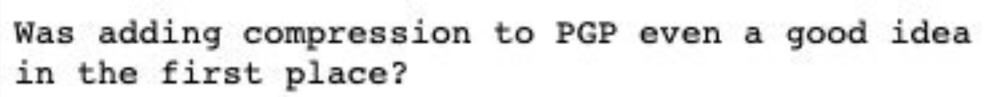
Peter



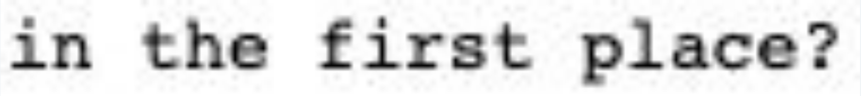
Date: Mon, 4 Jul 2022



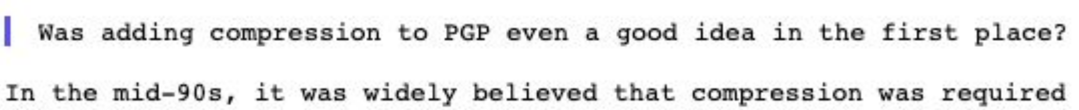
I don't think this one is impacted by max-output. Worse, I was told "Not a bug, sorry" by Werner.



Was adding compression to PGP even a good idea in the first place?

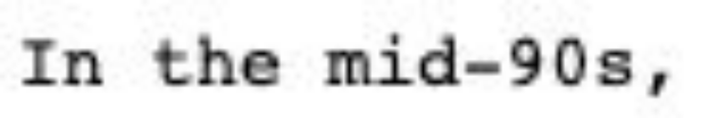


in the first place?



> Was adding compression to PGP even a good idea in the first place?

In the mid-90s, it was widely believed that compression was required

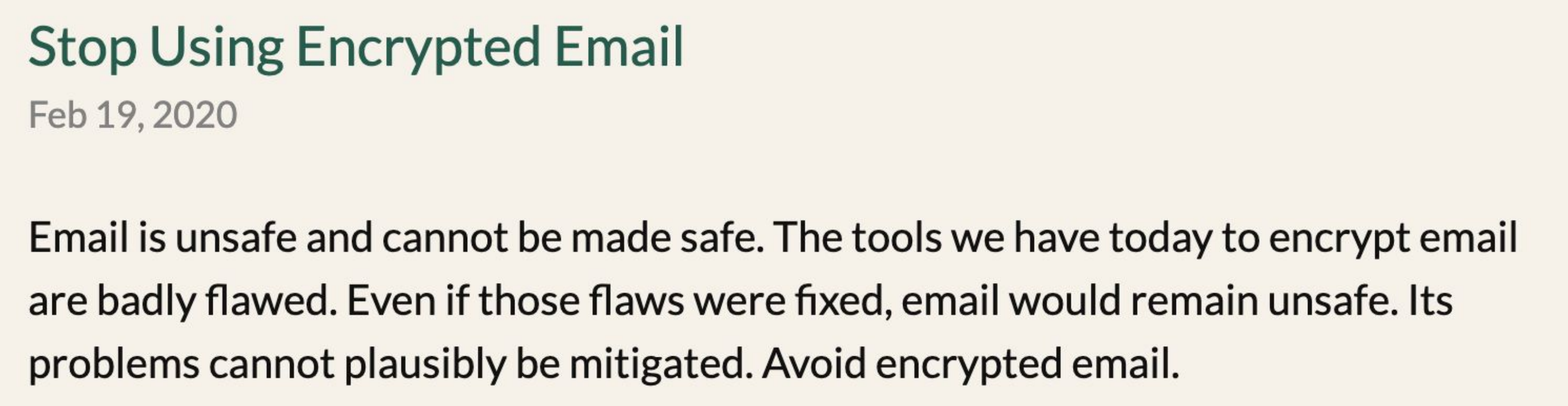


In the mid-90s,

PGP lures you into a false sense of security

even *if* you do everything right, PGP will not save you from:

- **unencrypted metadata (including your recipients and subject line)**
- **people helpfully replying in cleartext**
- **nonrepudiation (maybe you didn't mean to send that particular email with a permanent global identifier for yourself?)**



Stop Using Encrypted Email

Feb 19, 2020

Email is unsafe and cannot be made safe. The tools we have today to encrypt email are badly flawed. Even if those flaws were fixed, email would remain unsafe. Its problems cannot plausibly be mitigated. Avoid encrypted email.

“i can fix him”

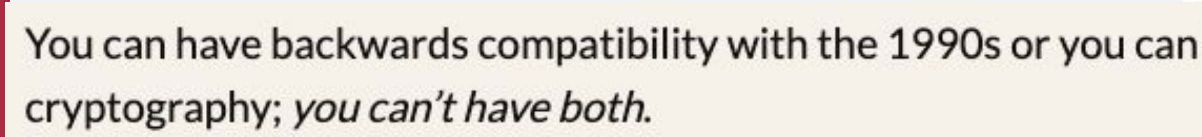
no, you can't

you have two options in the PGP ecosystem:

- accept (and generate) all kinds of crap required by the RFCs
- do your own thing and use nonstandard ciphers
  - in effect, a GnuPG monoculture. why not simply use something better to begin with?

we've only scratched the surface here. i'm going easy by not talking about the CLI.

JUST USE SOMETHING ELSE!!!!



You can have backwards compatibility with the 1990s or you can have sound cryptography; *you can't have both.*

what should you use?

texting friends/buying drugs? use Signal.

emailing anyone? stop LARPing*.

encrypting files? use age. online? use tarsnap.

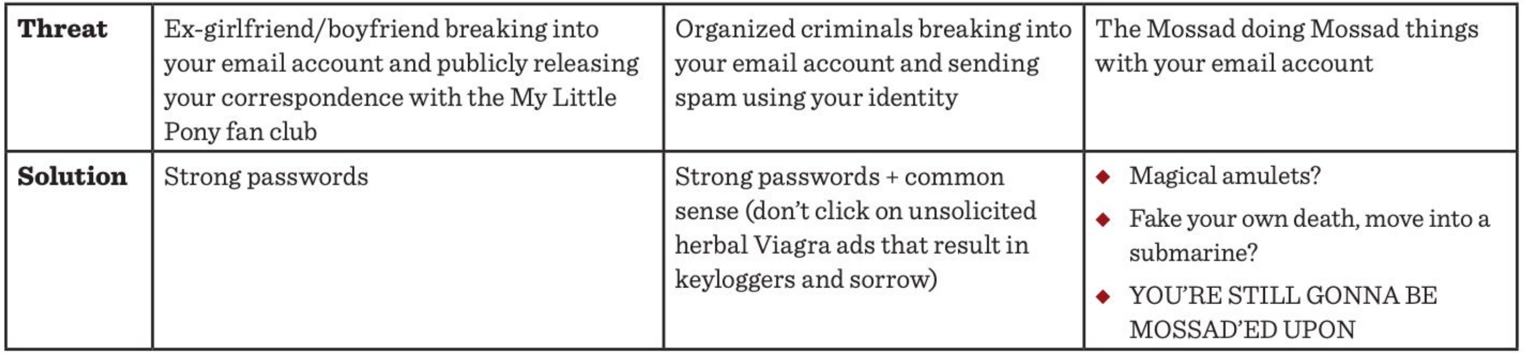
signing commits? use ssh (hurry up, GitHub)

age

FILE ENCRYPTION

Tarsnap

Online backups for the truly paranoid



Threat	Ex-girlfriend/boyfriend breaking into your email account and publicly releasing your correspondence with the My Little Pony fan club	Organized criminals breaking into your email account and sending spam using your identity	The Mossad doing Mossad things with your email account
Solution	Strong passwords	Strong passwords + common sense (don't click on unsolicited herbal Viagra ads that result in keyloggers and sorrow)	◆ Magical amulets? ◆ Fake your own death, move into a submarine? ◆ YOU'RE STILL GONNA BE MOSSAD'ED UPON

Figure 1: Threat models

the future: keyless codesigning with sigstore

when is the last time you verified a Python package's PGP signature?

when is the last time *you* published a PGP signature for your code?

- **hint: your linux package manager's signatures does not count**

sigstore is the future here:

- **signers use their *identities* (established through OIDC) instead of keys**
  - short lived signing certs + CT logging ensures auditability and transparency
- **Trail of Bits is working on this future**
  - sigstore-python + PyPI support for OIDC

conclusion: PGP delenda est

send me PGP hate mail:
william@yossarian.net

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFVkpc4BEACj1sixEQcfbaMqs4jeI2gdZ
PUetr1W8Yf2aBfmlIXUntvzpqxyj6l56YzL
XGJG/0a+UpN88ZE+H/G0HbkmDx9rqWJw0iRQs
wUxr4oncejLP9Fe/LCbdramD6jpFsT8SVXLco
5DGbaOYKTdIWyOMIoEZ5TpnIXHRP5jnvSiKJG
fq68V0jSAyw6E0kgjAYnxvdwp9h+xHUAYiegI
eKlgXis4qyZ8g/yR7o5T9oRafbNoGWY+fSPJM
8dy0oirescLRguPtAGeuFA1saOWY76AJuAF+d
Qu/46ClorhsrhybUmibbYPInNk0C9fR0uFVTw
9jUkF7/l3Pui7D7tS5iEoExGaw+OXo3UzT4G2
OyAFvdLg6PUQLlyouLyyusLbBoYPEs0YBdo5I
jeeqZ/S3GmWaIWOAV+b3bfQgLYVYCBOESEbCX
sq0G37KrgC3KV1w68k+lC6FvjHyIvmMftZ80B
cIvmCJa3i5kVxnQS4SdR7+vwmYEKkLXYMTs5G
hwwpi4Gipr9MdY9SyRSWLqpUxOHKdeB8d38vD
o6W/y1/uJ/GK3cfkLZL5AWvzoZKlILuSgKqlG
kRu2CYejVOMTj5A9gUMJ0hv/evlf7DTTQCXLA
JNrMSjU+youGA4gB4v2S132lrB8Pe3SAZzHZJ
TbyhCpiZN3fdn4kXcUu91gr7yu90q6zL/QARA
QABtCtrZXliYXNlLmlvL3lvc3NhcmlhbiA8eW
9zc2FyaWFuQGtleWJhc2UuaW8+iQI+BBMBAgA
oBQJVZKXOAhsjBQkSzAMABgsJCAcDAgYVCAIJ
CgsEFgIDAQIeAQIXgAAKCRCFrgDFBIM7PKtsE
ACYY0zRgQtw3wRyHn8WeEQ8lKUrv6V4hbzgi4
```

cribbed material

- “SKS Keyserver Network Under Attack”, Hansen, GitHub Gist 2019
- “Fuck RSA”, Ben Perez, SummerCon 2019
- “On the (In)security of ElGamal in OpenPGP”, Feo et al., RWC 2022
- “Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels”, Poddebniak et al., USENIX 2018
- “This World of Ours”, James Mickens, ;login January 2014
- “Stop Using Encrypted Email”, Latacora, February 2020
- “The PGP Problem”, Latacora, July 2019
- "Giving up on long term PGP", Filippo Valsorda, December 2016

thanks

- ryan stortz (@withzombies)
- josh hofing