NAVIGATION

CONTENTS

15 reasons not to start using PGP

Because of popular demand, here's the collection of reasons to prefer more advanced cryptographic communications tools and stop investing in the old PGP over e-mail architecture, the problem mostly being e-mail rather than PGP.

Pretty Good Privacy is better than no encryption at all, and being end-to-end it is also better than relying on SMTP over TLS (that is, point-to-point between the mail servers while the message is unencrypted in-between), but is it still a good choice for the future? Is it something we should recommend to people who are asking for better privacy today?

The text concludes mentioning some of the existing alternatives, so, again, this is *not* about not using encryption. It is about not falling into the intellectual trap of giving backwards compatibility the highest priority.

NAVIGATION
CONTENTS

1. Downgrade Attack: The risk of using it wrong.

With e-mail the risk always remains that somebody will send you sensitive information in cleartext - simply because they can, because it is easier, because they don't have your public key yet and don't bother to find out about it, or just by mistake. Maybe even because they know they can make you angry that way – and excuse themselves pretending incompetence. Some people even manage to reply unencrypted to an encrypted message, although PGP software should keep them from doing so.

The way you can simply not use encryption is also the number one problem with OTR, the off-the-record cryptography method for instant messaging.

This opens up for a great possibility for attack: It's enough to flip a bit in the communication between sender and recipient and they will experience decryption or verification errors. How high are the chances they will start to exchange the data in the clear rather than trying to hunt down the man in the middle?

The mere existence of an e-mail address in the process is a problem. Next generation cryptographic communication tools simply do not provide means to exchange messages without encryption, so if something goes wrong at least there is no doubt it could be you doing it wrong -- and giving up on privacy becomes at least a very conscious choice.

Update: And it's not like it's a problem only for the less careful or less tech-savvy. A notable cryptographer recently sent out confidential mail unencrypted. People told him, but he didn't believe it. He wrote himself encrypted mail and indeed, there it was, the mail in the clear. Turned out that one specific version of enigmail was in some strange way incompatible with a specific version of Thunderbird, sufficiently to pretend a completely

normal user experience, yet the mails would go out unencrypted, leaving just a remark somewhere in the messages log. There was no way even for the most experienced user to protect himself from a software attack of this kind. This can happen to you, too. Anytime you upgrade your operating system. But only with encryption-on-top systems like PGP.

2. The OpenPGP Format: You might aswell run around the city naked.

Thanks to its easily detectable OpenPGP Message Format it is an easy exercise for any manufacturer of Deep Packet Inspection hardware to offer a detection capability for PGP-encrypted messages anywhere in the flow of Internet communications, not only within SMTP. So by using PGP you are making yourself visible. Stf has been suggesting to use a non-detectable wrapping format.

Update: Gregory mentions that by using the –hidden-recipient flag you can tell PGP to, at least, hide who you are talking to. Hardly anyone does that: "PGP easily *undoes* the privacy that an anonymity network like Tor can provide" (by including the recipient's public key in the message).

Update 2015: Several new crypto systems for e-mail such as opmsg have surfaced since writing of this document. They address just this and a few other problems with PGP but still suffer from all other problems given by SMTP.

3. Transaction Data: Mallory knows who you are talking to.

Should Mallory not possess the private keys to your mail provider's TLS con-

nection yet, he can simply intercept the communication by means of a man-in-the-middle attack, using a valid fake certificate that he can make for himself on the fly. It's a bull run, you know?

Side note: Did you ever see a mail returned to you because of an invalid TLS certificate? And you can bet the net is full of invalid certificates. In most cases the mail will be delivered anyway, so Mallory doesn't even have to fake a valid certificate. He can use an invalid one, too.

Even if you employ PGP, Mallory can trace who you are talking to, when and how long. He can guess at what you are talking about, especially since some of you will put something meaningful in the unencrypted *Subject* header. PGP offers a means to encrypt the Subject line by now, but have you seen anyone use it?

Should Mallory have been distracted, he can still recover your mails by visiting your provider's server. Something to do with a PRISM, I heard. On top of that, TLS itself is being recklessly deployed without forward secrecy most of the time.

Update: This so-called *metadata* about who is talking to whom is of constitutional importance. It is a founding requirement of democracy to be able to share critical thinking and organize as a political group outside the view of government and not give anyone the power to influence, manipulate or keep a new democratic movement from growing and developing its potential. See the update below for more on this kind of reasoning.

4. No Forward Secrecy: It makes sense to collect it all.

As Eddie has told us, Mallory is keeping a complete collection of all PGP mails being sent over the Internet, just in case the necessary private keys

may one day fall into his hands. This makes sense because PGP lacks forward secrecy. The characteristic by which encryption keys are frequently refreshed, thus the private key matching the message is soon destroyed. Technically PGP is capable of refreshing subkeys, but it is so tedious, it is not being practiced – let alone being practiced the way it should be: at least daily.

Update 2015: At least two new crypto schemes over SMTP have been invented that implement forward secrecy but aren't PGP-compatible. One is called opmsg. The other one I forgot. They don't address most other problems mentioned here.

5. Cryptogeddon: Time to upgrade cryptography itself?

Mallory may also be awaiting the day when RSA cryptography will be cracked and all encrypted messages will be retroactively readable. Anyone who recorded as much PGP traffic as possible will one day gain strategic advantages out of that. According to Mr Alex Stamos that day may be closer than PGP advocates think as RSA cryptography may soon be cracked.

This might be true, or it may be counter-intelligence to scare people away from RSA into the arms of elleptic curve cryptography (ECC). A motivation to do so would have been to get people to use the curves recommended by the NIST, as they were created using magic numbers chosen without explanation by the NSA. No surprise they are suspected to be corrupted.

With both of these developments in mind, the alert cryptography activist scene seems now to converge on Curve25519, a variant of ECC whose parameters where elaborated mathematically. "They are the smallest numbers that satisfy all mathematical criteria that were set forth" explains Christian

Grothoff of GNUnet.

ECC also happens to be a faster and more compact encryption technique, which you should take as an incentive to increase the size of your encryption keys.

Unfortunately, thanks to RFC 6637 GnuPG now supports ECC with both Curve25519 and the suspicious NIST curves, but you can only activate those in ultra expert mode.

Nadia Heninger tells us some more on the topic, and concludes that there is no proof that mathematical discoveries cannot cause a cryptographic meltdown anytime: "Just because nothing has happened for two decades doesn't mean that something cannot happen." It is up to you to worry if it's more likely that RSA or ECC could be cracked in future. Should a mathematical breakthrough drop from the sky, probably both would be affected.

As a side note, OpenPGP requires the use of SHA1 for its fingerprinting. That means the way most people are authenticated in PGP may someday fall apart.

6. Federation: Get off the inter-server super-highway.

NSA officials have been reported saying that NSA does *not* keep track of all the peer-to-peer traffic as it is just large amounts of mostly irrelevant copyright infringement. It is thus a very good idea to develop a communications tool that embeds its ECC- encrypted information into plenty of P2P cover traffic.

Although this information is only given by hearsay, it is a reasonable consideration to make. By travelling the well-established and surveilled paths of

e-mail, PGP is unnecessarily superexposed. Would be much better, if the same PGP was being handed from computer to computer directly. Maybe even embedded into a picture, movie or piece of music using steganography.

Also, there are several issues about Federation itself… if all the people run their own servers instead of developing distributed serverless solutions, this is a guarantee that the cloud industry will always be several steps ahead.

7. Discovery: A Web of Trust you can't trust.

Mike Perry has made a nice collection of reasons why the PGP Web of Trust is suboptimal. It is in many ways specific to the PGP approach and not applicable to other social graphs like secushare's. Let's summarize: The PGP WoT

1. is publicly available for data mining,
2. has many single points of failure (social hubs with compromised keys) and
3. doesn't scale well to global use.

So these are actually three more reasons not to use PGP, but since you can use PGP without WoT we'll count them as one.

Update: Just found out that when you look up a key your amazing PGP client will by default do a cleartext HTTP request to the key server, so anyone can see who your conversation partners are.

8. PGP conflates non-repudiation and authentication.

"I send Bob an encrypted message that we should meet to discuss the suppression of free speech in our country. Bob obviously wants to be sure that the message is coming from me, but maybe Bob is a spy … and with PGP the only way the message can easily be authenticated as being from me is if I cryptographically sign the message, creating persistent evidence of my words not just to Bob but to Everyone!" *(Thanks, Gregory, for providing this one ;-))*.

OTR has introduced deniable authentication to address this problem and many next generation tools have adopted that concept. OTR cryptographically allows two people to be sure who they are talking to, yet they cannot prove it to anybody else.

9. Statistical Analysis: Guessing on the size of messages.

Especially for chats and remote computer administration it is known that the size and frequency of small encrypted snippets can be observed long enough to guess the contents. This is a problem with SSH and OTR more than with PGP, but also PGP would be smarter if the messages were padded to certain standard sizes, making them look all uniform.

10. Workflow: Group messaging with PGP is impractical.

Have you tried making a mailing list with people sharing private messages? It's a cumbersome configuration procedure and inefficient since each copy is re-encrypted. You can alternatively all share the same key, but that's a different cumbersome configuration procedure.

Next generation communication tools automate the creation and distribu-

tion of group session keys so you don't need to worry. You just open up a working group and invite the people to work with. It's so simple, people worry it may not be happening.

11. Complexity: Storing a draft in clear text on the server

Update: These days mail tools are too complicated. Here come enigmail that is in charge of encrypting mails before they leave Thunderbird. But wait, didn't Thunderbird just store a draft? Yes, and since I happen to have IMAP configured it stored the draft to my server. Did it bother that I had checked the flag that I intend to encrypt the mail? No, the draft is on the server in the clear. I look around and find out that Claws has been having the same bug. I'm not surprised, after all it's the most natural way of doing things. One person implements IMAP, another implements PGP support, and they never bump into each other and realise that the default behaviour of a mail agent that supports both is to do what it should in no way ever do: send the unencrypted mail to the server. This makes the entire effort to use PGP useless. I looked around for warnings, but even the best manuals for doing PGP correctly are aware of a lot of problems, but not this one. I am only on day three of *really* using PGP, and I already discovered a security flaw that no-one has talked about much ever before. Is this normal? I have Thunderbird 17.0.8 and you?

P.S. I recommend you to turn off saving mail drafts to the server.

12. Overhead: DNS and X.509 require so much work.

This may seem unrelated, but PGP builds upon e-mail, and e-mail unnecessarily enforces a dependency on DNS and X.509 on us (the TLS and HTTPS

certification standard that makes us need certificates, signed by an *authority*, and then can be fooled and broken anyway). Both cost money to participate in and have to be meticulously administered. Anyone who tried to do it, knows: Mail (and also Jabber) server administration is annoying and expensive.

Next generation alternatives are either based on DHT technology, social graph discovery, byzantine consensus (aka blockchain) or opportunistic broadcast. All of them are powered by the mere fact that you are using the software. Frequently there will be sponsored servers providing for faster service, as it has become the standard for Tor, but the administration of such servers is trivial: Just unpack the software and run it (exit nodes are a special case which are only relevant if you care to access the legacy broken Internet).

Why are you accepting being enslaved by e-mail?

13. Targeted attacks against PGP key ids are possible

PGP has a bad habit of using truncated fingerprints as key ids, organizing keys in its database by short key id and dealing keys with the same short key id as probably being the same, although it isn't so hard to make a new key pair that resolves to the same key id as an existing one. This seems to be a problem even with long key ids. Now people say you should use the full fingerprint, but I remember a time when it was said that the purpose of fingerprints is just for simplifying the comparison of keys among human beings. Computers should *always* ensure the identity of a public key by comparing nothing less than the *complete* public key. By using short ids for maintaining keys the PGP software implementations are doing it wrong.

NAVIGATION
CONTENTS

One possible consequence of this is that users could be tricked into accepting a false replacement key from a key server or in some other way confuse their key management to the point of corrupting a communication path that used to be safe and allowing a man in the middle into the game. People who have just their short key id printed on their business card could suffer targeted man in the middle attacks: The MITM just needs to intercept the keyserver look-up, which as we know is unencrypted by default, and produce the false recipient data. The MITM must then also intercept in- and outgoing SMTP traffic in order to re-encrypt the mail conversation on the fly to the actual key the recipient expects and vice versa. This can in fact be automated to undermine the PGP infrastructure on a large scale, but it would not go unnoticed whereas a targeted attack most likely would.

You can make the attack slightly more difficult by using encrypted key server look-ups (= learn to configure gpg to use sane defaults), but since the key servers do not use PGP to authenticate themselves you can still suffer a MITM attack on the TLS certification level (see X.509 above). And of course there is also the possibility of the key server itself being used in a targeted operation against you. In practice the only currently secure way to communicate a key on a business card is to print its entire fingerprint along with the look-up id – and not forget to actually check it (happened to me, so I bet it happens to you).

Update: Apparently this problem has been addressed in GnuPG 2.1. Users that refuse to publish their keys to a keyserver in a desperate attempt to protect their metadata may however still be subject to confusion by an imposter that posts a key with the same long id.

14. TL;DR: I don't care. I've got nothing to hide.

So you think PGP is enough for you since you aren't saying anything *reeaally* confidential? Nobody actually cares how much you like to lie to yourself stating you have nothing to hide. If that was the case, why don't you do it on the street, as John Lennon used to ask?

It's not about you, it's about your civic duty not to be a member of a predictable populace. If somebody is able to know all your preferences, habits and political views, you are causing damage to democratic society. That's why it is not enough that you are covering naughty parts of yourself with a bit of PGP, if all the rest of it is still in the nude. Start feeling guilty. Now.

It's also about your entire social environment. Your friends, your family deserves better than to end up in XKEYSCORE. You have no *right* to waive away *their* privacy. Each time you log in into Facebook or Whatsapp you are committing a felony against *them*.

Update: Read about the fallacy of transparency.

15. The Bootstrap Fallacy: But my friends already have e-mail!

But everyone I know already *has* e-mail, so it is much easier to teach them to use PGP. Why would I want to teach them a new software!?

That's a fallacy. Truth is, all people that want to start improving their privacy have to install new software. Be it on top of super-surveilled e-mail or safely independent from it. In any case you will have to make a safe exchange of the public keys, and e-mail won't be very helpful at that. In fact you make it easy for Mallory to connect your identity to your public key for all future times.

So installing a brand new software that only provides for safe encrypted communications is actually a less complicated change of habits than trying to fix the e-mail system, then learning how to use PGP without messing it up.

If you *really* think your e-mail consumption set-up is so amazing and you absolutely don't want to start all over with a completely different kind of software, look out for tools that let you use mail clients on top - not the other way around. Bitmessage has an IMAP emulation for example.

But what should I do then!??

So now that we know *n* reasons not to use e-mail and PGP, let's first acknowledge that there is no obvious alternative. Electronic privacy is a crime zone with blood freshly spilled all over. None of the existing tools are fully good enough. We have to get used to the fact that relevant new tools will come out all the time, and you will want to switch to a new software twice a year. Mallory has an interest in making us believe encryption isn't going to work anyway – but internal data leaked by Mr Snowden confirms that encryption actually works. We should just care to use it the best way.

There is no one magic bullet you can learn about.

You have to get used to learning new software frequently. You have to teach the basics of encryption independently from any software.

In the comparison we have listed a few currently existing technologies that provide a safer messaging experience than PGP. The problem with those frequently is, that they haven't been peer reviewed. You may want to invest

NAVIGATION
CONTENTS

time or money in getting projects reviewed for safety.

Thank you, PGP.

Thank you Mr Zimmermann for bringing encryption technology to the simple people, back in 1991. It has been an invaluable tool for twenty years, we will never forget. But it is overdue to move on.

Questions and Answers

Some questions were posed on libtech which deserve an answer:

What's the threat model here?

What if Mallory isn't a well-funded governmental organization but is the admin who runs your employer's email servers?

That's a good point. The reason why I don't pay attention to lesser threat models is that the loss in quality of democracy we are currently experiencing is large enough that I don't see much use for a distinction of threat models - especially since alternatives that work better than PGP exist, so they are obviously also better for lesser threat models.

For example, I don't think that a dissident in Irya (ficticious country) is better off if no-one but Google Mail knows that they are a dissident. Should at any later time in their life someone with access to that data find it useful to use it against them, they will. And who knows what the world looks like in twenty years from now?

Not saying give up and die. Saying if you can opt for better security, don't postpone learning about it. If you can invest money in making it a safe option, don't waste time with yet another PGP GUI project or the crowdfunding hype of the day.

If employers, schools, parents, skiddies can find out who you are exchanging encrypted messages with, that can be a very real threat to you. Using a tool that looks like it does something totally different.. on your screen, over the network and even on your hard disk.. can save your physical integrity.

Is this about PGP or rather about e-mail?

It's more about SMTP, but I don't think it makes much difference for the end user whether SMTP federation or actual PGP is failing them.

What about S/MIME?

"S/MIME unfortunately suffers from many of the same issues as OpenPGP, and then some more." I don't find S/MIME worth mentioning anymore. It is based on X.509 which has so failed us.

We need a new open standard first!

Open standards are part of the problem, not the solution. It is a **very bad** development that it has become en vogue to require standardization from projects that haven't even started functioning. It has been detrimental to the social tool scene: None of them work well enough to actually scale and

replace Facebook, but the scalability problems are already being cemented into "open standards," ensuring that they never will function. Same thing happened with Jabber as it turned into XMPP.

You must *always* have a working pioneer tool *first*, then dissect the way it works and derive a standard out of it. Bittorrent is a good example for that. It's one of the few things that actually works. Imagine if Napster and Soulseek had developed an open standard. It would only have delayed the introduction of Bittorrent, promoting an inferior technology by standardization. Another good example is Tor – it was able to improve each time somebody figured out a way to attack it, because it didn't have a long-term legacy compatibility requirement like SMTP, DNS or XMPP.

Why don't we fix all of these problems with PGP and e-mail?

Even if all the effort is done that a project like LEAP is striving for, you will still be receiving SPAM and unencrypted mail, just because you have a mail address. You will still have a multitude of hosts that are still "unfixed" because they don't care to upgrade. You will still carry a dependency on DNS and X.509 around your neck just to be able to be backwards compatible to an e-mail system of which you hope you won't have to send or receive any messages since they will damage your privacy. And I still don't see by which criteria a dissident should pick a trustworthy server. I know I can rent one, but even if I have a root shell on my "own" server, it doesn't mean it is safe. It's better not to need any!

So what is this terrific effort to stay backward compatible good for? I don't see it being a worthwhile goal. There is so much broken about it while a fresh start, where every participant is safe by definition, is so much more useful. Especially you don't have that usability challenge of having to ex-

NAVIGATION
CONTENTS

plain to your users that some addresses are superduper safe while other addresses are lacking solid degree of privacy.

One major problem with the new generation of privacy tools is, they are *so simple*, people have a hard time believing they are actually working.

TOP