p≡p

Privacy by Default.

White Paper

p≡p foundation council

2016–07–18 (v1.0.5)

Contact: mailto:council@pep.foundation[1]
Website: https://pep.foundation[2]

[1] OpenPGP fingerprint: EC55 39C8 FECF 7C4F 324B F027 A9DE 30FC 56BB B555

[2] Or using CAcert certificate cf. URL https://cacert.pep.foundation

Document versions

- v1.0.5 (2016-07-18): Update modules graphics (pEp for iOS; no Inboxcube patch)
- v1.0.4 (2016-07-02): Change OpenPGP fingerprint in the title page
- v1.0.3 (2016-03-03): Rename "Conceptual White Paper" in "White Paper"
- v1.0.2 (2016-03-02): One typographical fix
- v1.0.1 (2016-02-29): Stylistic changes and typographical fixes
- v1.0 (2016-02-29): First public version
- v0.6 (2016-02-29): Draft version
- v0.5 (2016-02-28): Second revised early draft
- v0.4 (2016-02-21): First early draft

Contents

1 Introduction and Motivation

The Right to Privacy and the Right to Freedom of Information are part of the inalienable Human Rights.

These rights are listed in Article 12 and Article 19 of the Universal Declaration of Human Rights based on Resolution 217 A (III) of the UN General Assembly as Objects of Legal Protection.

In a time when Chartered Rights are being more and more threatened, measures have to be taken into our own hands in order to protect and preserve them. Especially because of the Internet the circumstances for a majority of people have changed completely, and unfortunately not for the better.

Taking a deep breath, we must realize that the entrance of humanity into digital age is accompanied by all of us continuously losing more and more of our privacy rights, which we believe are paramount for citizens, families, enterprises, public offices, lawyers, NGOs, journalists and political activists for being able not just to feel, but also to act and interact freely – without any fears of becoming discriminated or suppressed based on past behaviors, opinions expressed or the social communication network an actor is in.

Today, any individual or collective actor engaged in any digital activity, must assume not only to be just constantly monitored, but having its behavior automatically analyzed by text and data mining or other technologies of machine learning[3] – leading for literally everyone being scored and categorized. This generally happens without the actors' consent and without a clear consciousness of the whereabouts of all the personal, collective or corporate data involved.

In the future, mass deployment of Internet of Things (IoT) technologies is expected, meaning that many of the objects we know today – may that be implants in our bodies, (parts of) our clothes, all kind of sensor systems in our homes, streets or organizations – will get "on-line" and "talk" to each other. Such developments have the potential to completely eradicate the notion of privacy. This puts in danger an open and democratic society.

All that said, it's vital to act now and in the most effective way possible: p≡p stands for pretty Easy privacy (abbreviated as p≡p) – the "E" or "≡" to be emphasized, but without compromise in privacy. We want the Internet to become a secure place, where people and organizations gain back their constitutional rights to communicate in private by default.

We start our mission by radically easing the use of well-known and established end-to-end cryptographic tools for already existing and widely used written digital communication channels (like e-mail, SMS or chat), without forcing users to adhere to new communications channels and by providing the necessary tools for gluing already existing – but not easy to use – components together and helping businesses to spread p≡p and the technologies it employs as industry standards.

However, considering the different use cases the Internet has today (e. g. it's also used for voice calls or to carry

[3] Sometimes also just called "big data" technologies involving (complex) algorithmic methods.

out payments), we don’t want p≡p to be seen as forever bound and ending in “just” securing written digital communciations: at least, we let that be the beginning of our efforts to help to secure the Internet and its users.

2 p≡p at a glance: privacy without compromise

What is p≡p?

p≡p motivates a new standard to securely encrypt and verify written communications, without reinventing the wheel. p≡p, by design, eases secure communications relying on well-established end-to-end cryptographic methods (following standards like OpenPGP or OTR) by integrating into existing systems for written digital communications and automating key management tasks.

Ultimately, p≡p wants to change the default in written digital communications: from unencrypted, unverified and unanonymized to encrypted, verifiable and anonymized.[4]

p≡p restores security and privacy without requiring user interaction for its basic operation and never putting communications' security over the ability to communicate: thus, if no secure communication channel between two peers can be established, the text message is sent unencrypted (cf. 4 for some more details) rather than hassling the users involved.

Whenever two p≡p peers established a technically secure communication channel, they can add human trust to it by verifying each other's digital identity by comparing *Trustwords* (cf. 3.2 for some more details): like that two peers can increase confidence they are really communicating with each other and no Man-In-The-Middle (MITM) attack is taking place. No trust will be put into key servers or centralized trust infrastructures (like commercial certificate authorities) by default, even though p≡p (for compatibility reasons and integration purposes) supports S/MIME (e. g. used in organizations for e-mail encryption). However, default p≡p installations shall not be expected to support non-end-to-end approaches of cryptography: for p≡p centralized cryptographic approaches provide weak cryptography only and are considered an *unreliable* communication type in terms of our approach of privacy without compromise.

To achieve the goal of spread, from an organizational point of view, the p≡p project founders set up two different types of entities, distinguished by separated legal entities in two different European countries:

- p≡p foundation, a non-profit organization established as foundation under Swiss law holding the trademarks on p≡p and the copyrights of the p≡p engine and the p≡p adapters for different programming languages and development environments. It is supporting Free and Open Source Software Projects (FOSS) to integrate p≡p.
- p≡p security, a enterprise incorporated in Switzerland and Luxembourg and licensed by p≡p foundation to spread p≡p software to both Business-to-Business (B2B) and Business-To-Consumer (B2C) markets by creating, distributing and providing support for add-ons, plug-ins and apps enabling users and organizations to use p≡p technology.[5]

[4] For anonymization, GNUnet message transport is considered, cf. `https://www.gnunet.org`.
[5] Cf. `https://www.prettyeasyprivacy.com/`

All software published by p≡p foundation will be and forever remain freely available for the general public under the GNU General Public License version 3 (GPLv3) or any later version. All software will be subject to an independent code review for any release, disclosing the full technical report. As part of its its model of security by design providing the most trust possible, p≡p foundation will make sure that all licensed p≡p software gets code-reviewed by independent entities for each public release. By all of these measures we do our best effort to make sure p≡p software is and stays backdoor-free even if it's distributed by third party.

p≡p software will be cryptographically signed and provided with detailed build instructions and a fully documented build chain for each platform involved, to make transparent how deterministic builds[6] can be created from source.

[6] Cf. https://blog.torproject.org/blog/deterministic-builds-part-one-cyberwar-and-global-compromise

3 Design principles

Above all, p≡p – contrary to existing cryptographic solutions – shall be easy to install, use and understand.

In our opinion, not just power users, but all users have the right to use end-to-end cryptography.

Furthermore, for their communications p≡p users don't depend on any specific (web or operating system) platform, message transport system (SMS, E-Mail, XMPP etc.) or centrally provided client–server or "cloud" infrastructures: p≡p is 100% peer-to-peer by design.

3.1 Ease of use: hassle-free installation, automatic configuration and integration

p≡p software shall integrate into existing systems for written digital communications like e-mail clients (e. g. Thunderbird, Apple Mail or Outlook) and messaging clients, or be provided as apps for smartphones (e. g. for Android or iOS) in a hassle-free way: in case of e-mail this means that upon installation existing OpenPGP keys will be used by p≡p automatically for any further communication; should no OpenPGP keys be available, p≡p automatically creates a private and public key for the e-mail address concerned. Even though it's discouraged for regular users, of course power users will remain free to configure their installation by themselves and turn off the automation proposed by p≡p.

From the beginning, the p≡p concept bears in mind to be installed not just on end user devices, but also in organizations with thousands of users: important features for such settings, like distribution lists, BCC mailings and unencrypted archiving of e-mails, which were sent or received encrypted, is possible.

3.2 Understandability: traffic lights semantics / Privacy Status

To indicate the security level in place for a given communication situation, p≡p uses traffic lights semantics[7] to easily indicate the *Privacy Status* to the user:

- **red**: An attack was detected or the user explicitly expressed mistrust for the given communication channel, so that this channel must be considered specifically insecure.
- **yellow**: The communication channel is perfectly encrypted by state-of-the-art technology known being correct, using well-implemented cryptographic methods only and sufficient key lengths, but sender (inbound) or recipient (outbound) are not (yet) verified.
- **green**: Two peers have used a side-channel (e. g. by making a phone call)

[7] Changes to the colors' meanings might still occur.

[8] *Trustwords* are unique representations of a peer's public key fingerprint, but not in form of hexadecimal blocks as usual when verifying OpenPGP fingerprints, but in form of words in users' mother native lan-

to check their *Trustwords*,[8] so the communication can be shown as fully trusted by all reasonable means.[9]

- No color: That's the default situation of today without p≡p or any other (known) cryptographic solution under management of p≡p software from an inbound or to an outbound peer. In case of e-mail the communication channel usually must be considered as insecure like sending a post card. Any important actor transporting the message can read about the sender, the recipients, the subject and read the message's full contents and attachments. Also in cases where the encryption employed is *undefined* or *unreliable* (e. g. S/MIME with commercial CAs) or a key couldn't be found the *Privacy Status* has no color.

3.3 Interoperability: convergence of messaging systems

Instead of forcing whole user bases to use new (post e-mail) messaging systems, the p≡p concept is about securing what's insecure today.

Not at last taking into account that e-mail is one of the most widespread message transport systems used[10] for written digital communication, sided by SMS, XMPP and proprietary message formats.

p≡p pursues the philosophy not to take sides, respecting users' choices and providing solutions for their preferred messaging systems and the ability to communicate with all peers on their preferred communications channels. Even though p≡p will provide its own messaging format for cases where both communication end points have p≡p software installed, and no other format delivers the needed properties.

3.4 Cross-platform support: free choice of platforms

From the very beginning (cf. 5 for some more details) portability and cross-platform support is an important design aspect of p≡p. p≡p will support desktop, apps and web-based systems. For the latter, add-ons and plug-ins will be made available supporting popular web-based e-mail providers, which people are being used to access with their different devices by a browser.

p≡p is starting with securing e-mail communications for Mail User Agents (MUAs) like Thunderbird, Apple Mail and Outlook providing add-ons, plug-ins and apps for smartphone operating systems like Android and iOS. p≡p shall also be integrated and shipped with already popular messaging systems wherever possible.

guage or any of the languages in the realm of ISO 639-1 code; cf. URL `https://cacert.pep.foundation/trac/wiki/Trustwords`.

[9] This, of course, is assuming no malware of any kind is present at one or both end points and no backdoor is present at any end point in the system.

[10] That is including the fact each active Internet user might have at least one e-mail address and the use of most (web) services require an active e-mail address to be used. Also in business situations, e-mail is persisting being prevalent.

3.5 100% peer-to-peer: no central providers

By default, communications between p≡p peers always work end-to-end encrypted – no eavesdropping in between shall be possible by design. p≡p users are never forced to use a proprietary platform.

Concerning communication by e-mail using OpenPGP cryptography, this means every user or organization can decide by themselves if they want to run an own e-mail server or to use third party services to communicate with other peers via e-mail.

In any case, p≡p will do its best to hide as much as possible of the communicated information from attackers – including metadata – increasing attackers' costs to the maximum (cf. 4 for more details).

p≡p delivers a synchronization protocol[11] allowing a user to own different devices (e. g. a laptop and a smartphone) sharing a common *Device Group* key pair, such that messages can be sent and read across different devices.

[11] Cf. URL `https://cacert.pep.foundation/trac/wiki/p%E2%89%A1p%20sync%20protocol` for a sketch.

4 Cryptography being used and communication strategy

Following the principle of privacy without compromise p≡p is making sure the p≡p engine always chooses the cheapest available and most secure way of communicating possible, that is choosing the most feasible communication route, which at the same time is the most expensive for attackers.

The current communication strategy looks like the following:

1. When two p≡p users are communicating:

   a. if on-line communication available: OTR (or Axolotl) through GNUnet

   b. if on-line communication not available:

      i. if anonymizing platform available, p≡p static encryption based on OpenPGP through anonymizing platform (e.g. Qabel[12])

      ii. if anonymizing platform not available, fallback to p≡p static encryption based on OpenPGP

2. when a p≡p user is communicating with a non-p≡p user then depending on the capabilities of the non-p≡p user:

   a. if anonymizing and forward secrecy is possible, use that (i. e. OTR over GNUnet)

   b. if anonymizing but no forward secrecy is possible, use that (i. e. OpenPGP over Qabel)

   c. if forward secrecy is possible, use that (i. e. OTR)

   d. if hard cryptography but no forward secrecy is possible, use that (i. e. OpenPGP)

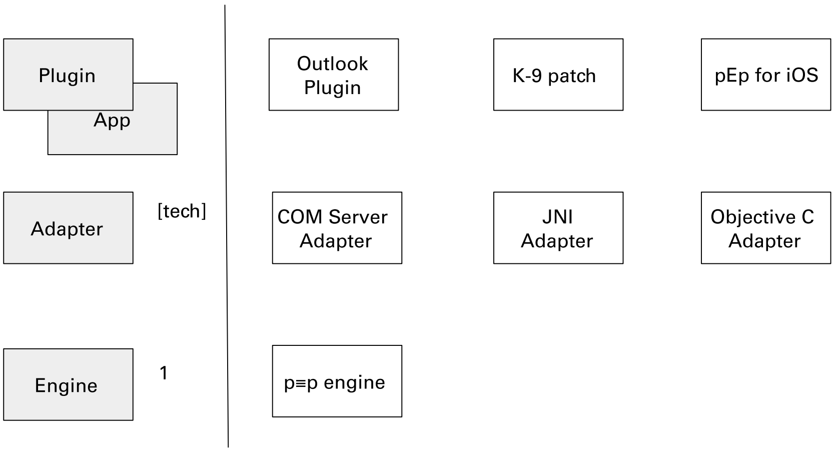
   e. if only weak cryptography is possible, use that (i. e. S/MIME with commercial CAs)

   f. send unencrypted

[12] Cf. URL https://qabel.de/ or any similar solution.

5 Architectural aspects: p≡p engine and adapters

p≡p global architecture



p≡p consists[13] of three components:

- p≡p engine
- p≡p adapters
- p≡p plug-ins, add-ons and apps

For portability reasons[14], the p≡p engine is written in C99 programming language. p≡p engine is implementing formats, applying cryptography, managing keys and trust and driving message transports.

Part of the p≡p engine distribution is a replacement for GnuPG, NetPGPet[15], a PGP implementation for platforms where GnuPG is not available.

Any developer of add-ons, plug-ins, apps or desktop applications does not need to deal with cryptographic functionalities accessible by the p≡p engine; instead p≡p adapters are providing an easy API in the language and the development environment of the application programmer.

Examples of adapters, which are already available under the GNU GPL v3 license by the p≡p foundation include:

- p≡p Qt Adapter for graphical environments based on Qt (e. g. KDE desktop systems)[16]
- p≡p Java Native Interface (JNI) adapter (e. g. for Android platforms)[17]
- p≡p iOS adapter for iOS-based systems (iPhones and iPads)[18]
- p≡p COM server adapter for Microsoft Windows, supporting a .NET API (e. g. needed for Outlook)[19]

More p≡p adapters to follow.

[13] For the most detailed, complete and up-to-date (work-in-progress) resource in this regard, the publicly available trac site of the p≡p foundation should be consulted: cf. URL https://cacert.pep.foundation/trac/.

[14] p≡p can also be made available for microcontrollers (on IoT components) with only minimalistic hardware.

[15] Cf. https://cacert.pep-project.org/trac/browser/netpgp-et/

[16] Cf. URL https://cacert.pep.foundation/trac/browser/pepqtadapter

[17] Cf. URL https://cacert.pep.foundation/trac/browser/pepjniadapter

[18] Cf. URL https://cacert.pep.foundation/trac/browser/pepiosadapter

[19] Cf. URL https://cacert.pep.foundation/trac/browser/pepcomserveradapter