StartMail Technical White Paper

Document owner Alex van Eesteren

Version 1.0.7

Date 2016-09-20

Table of Contents

1. Introduction

In recent years, and in 2013 in particular, a series of events have shown that even our everyday communications are valuable to eavesdroppers. Email is one of the most privacy sensitive services people use online, and keeping communications private is no simple task. Powerful standards such as SSL/TLS and OpenPGP already exist, but using them and setting them up correctly can present difficulties for many novice users. Due to the learning curve associated with setting up and using encryption, many people continue to communicate through insecure platforms.

StartMail aims to solve this problem. We consider privacy a fundamental human right, and we set out to bring privacy and security to the average user. StartMail was designed to help people regain their right to privacy by using powerful cryptography for their daily email communications.

StartMail brings this same ease of use to businesses that need to protect their information and comply with privacy regulations, like HIPAA. We explain in a separate section how we have made PGP business friendly by overcoming traditional PGP usability issues.

StartMail was created by the people behind the private search engines StartPage.com and Ixquick.com. Development took more than three years, and it was a natural step from offering private online search to offering secure and private email communications.

This document explains the choices we made when it comes to security practices and protecting the privacy of our users. We first give a quick overview of our services, and then we list the most important design choices that were made while designing our service architecture. Finally, we describe those decisions in detail, focusing on security-oriented precautions and features.

2. Overview of StartMail

StartMail is a platform for secure email communications. Our service can be accessed from a webmail interface, as well as through the traditional IMAP protocol, for compatibility with existing email clients.

Advanced functionality is available to power users who have enough knowledge and experience to benefit from it. For example, users can autonomously store their recovery code, import and manage existing OpenPGP keys, or even manually handle all the OpenPGP interactions altogether, making StartMail the relay platform for their own secure communications.

We aim to make secure communications as transparent as possible. To this end, OpenPGP is used. Power users have the option to opt out from all cryptography-related functionality and handle their cryptography themselves, then access their email through IMAP. But by default, StartMail offers the following security features to users:

- Asymmetric encryption and signing using OpenPGP to both StartMail and non-StartMail users
- Symmetrical (Question & Answer) encryption to users who do not use OpenPGP
- Company operations and legal entity based strictly outside of US jurisdiction

- Storage of a minimal amount of data about users and no tracking cookies. See our very strict privacy policy [1]
- All data belonging to users, including mail, preferences, and anti-spam profile, is stored encrypted in their "User Vault"
- Emails that arrive for users are immediately encrypted with the public key of their "queue key pair" for later delivery when the User Vault is opened by the user

3. Design Considerations

When creating StartMail, we were presented with several choices concerning security, privacy and user experience in our service. In this section we address the most notable decisions.

3.1. Webmail vs Desktop Client

Our goal in creating StartMail was to develop a beginner-friendly OpenPGP client. We decided to offer a webmail client rather than a desktop (or mobile) application for several reasons. First, many email users have grown accustomed to using a browser to access their mail. Second, since users expect to be able to access their mail from different devices, a webmail solution gives them an alternative to OS-specific applications and allows them to benefit from the ubiquity that browsers offer. Finally, there are already secure OpenPGP compatible desktop clients that can be configured to work with StartMail via IMAP.

The StartMail Web application is a PGP-enabled mail client for the Web. Nevertheless, we also offer full support for traditional email clients using IMAP.

3.2. Client-side vs Server-side Encryption

By design, OpenPGP operations (such as encryption and decryption) can take place either on the server or on the client. In StartMail, all OpenPGP operations take place server-side.

We have opted to perform cryptography on the server after thoroughly considering the client-side option. We rejected it because OpenPGP operations in a Web browser take place in a JavaScript context, which is not at all the right environment for cryptography. A number of compelling reasons why this is the case are described by NCC Group in this excellent article: https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2011/august/javascript-cryptography-considered-harmful/.

Among the reasons for rejecting client-side cryptographic operations are:

- Browser JavaScript is not ready for cryptography in terms of programming primitives, such as a reliable source of random numbers, mathematical functions etc.
- The malleability of the JavaScript runtime environment means that auditing the future security of a piece of JavaScript code is impossible: The server providing the JavaScript could easily place a backdoor in the code, or the code could be modified at runtime through another script. This requires users to place the same measure of trust in the server providing the JavaScript as they would need to do with server-side handling of cryptography.

- JavaScript is executed in an environment (the browser) over which the programmer has extremely little control. In these conditions it becomes hard or impossible to perform secure memory management, protect against timing attacks, and so forth.

In simpler terms: JavaScript is a poor environment for handling such a delicate operation as cryptography. What's more, JavaScript code that runs in the browser can be influenced by any other piece of JavaScript code running in the same browser window, and sometimes even by other pages. These conditions would make it impossible for us to provide the same level of security we can offer by delivering server-side cryptography.

A commonly cited benefit of performing client-side cryptography is that, in theory at least, the server or mail provider never has access to the user's OpenPGP private key, thus reducing the amount of trust that the user needs to place in the mail provider. In practice, however, this security benefit is illusory. The only way in which OpenPGP private keys are truly not in contact with any server or server-provided code is by performing cryptography through native desktop applications (e.g. GnuPG or GPGtools). StartMail already fully supports this through IMAP.

Client-side cryptographic operations performed in a Web browser using JavaScript offer only superficial protection against a server that has been compromised or has malicious intent. This is because the JavaScript code that is executed is received directly from the server. In this way, the browser becomes an extension of the server-side application for certain operations.

We view the client-side (browser) vs server-side debate as a moving target, and we will be watching for technological advancements that will allow us to revisit our decision. Especially in the field of browser-based cryptography, the availability and the quality of libraries change rapidly. We will consider implementing a client-side solution once the browser ecosystem and available libraries have developed to a degree necessary to provide trustworthy cryptographic operations.

There are trust issues involved regardless of whether OpenPGP operations take place in the browser or on the server. For this reason we believe "real client side" (native desktop) to be the most secure option. It allows users to have complete control over their OpenPGP key management and cryptography operations in an environment that is more suitable for that purpose than the browser. There are still obvious issues with user experience, setup complexity, and potential data loss in desktop clients. While this option is fully compatible by connecting to StartMail via IMAP, we also aim to offer the most secure browser-based solution we can.

3.3. Client-side vs Server-side Key Storage

Regardless of where encryption and decryption operations take place, a discussion about OpenPGP key storage is in order. Client-side key storage means that the OpenPGP private key is permanently stored on the users' computers (as opposed to being stored on the server). Users will either store the key permanently in the browser or they will carry their key with them and make it available to the browser when needed.

Browsers do not have a secure way of storing OpenPGP keys that meets our standards. What's more, the great majority of people still run outdated browsers. In other contexts, this could be solved with "graceful degradation" [2] (e.g. reverting to an alternative CSS property because the browser doesn't support newer ones). This is simply not a viable option in a

security model. Until this situation improves, we prefer not to expose our users' PGP keys to the dangers of in-browser storage. Since we chose the server-side approach for handling cryptographic operations (for reasons described above), we do not see a compelling case for exposing the key to browsers.

Needless to say, users carrying their OpenPGP keys around with them is a potential security concern. Users may not have the knowledge or patience to take the necessary precautions to ensure their key is not overly duplicated or stored insecurely.

3.4. Performance and Debugging Information vs Privacy

Keeping a log of activities that occur on our servers can be useful for debugging purposes, performance improvements and security enhancements. However, in deciding what data to log, we always tip the scale in favor of our users' privacy and security.

A key example of this is our approach to handling spam filter profile data. Along with the standard algorithms, we use metadata to obtain real-life examples of what users do and do not consider spam. We choose to store this information in the user's personal User Vault (described below), instead of using it generally to enhance the overall effectiveness of our spam filter. To do otherwise would mean making parts of a user's emails available within the whole system, which would violate our commitment to making privacy our top priority.

A similar situation exists for search indexing. In order for the search functionality to respond quickly, email needs to be indexed and indices must be up to date. This can happen at any point in time, provided it occurs before the user begins a search. In the case of StartMail, however, the user's search indices can only be updated when the user is logged in. When the user is not logged in, emails and indices are stored encrypted in the User Vault, and indexing operations cannot take place. The constraint that forces us to index email only upon log-in provides better privacy and security for our users at the expense of a slight drawback in terms of processing speed.

3.5. Open Source vs Closed Source Software

StartMail's source code consists of a mix of open-source and closed-source components. The open-source components are mainly infrastructure-related code (such as the Linux operating system and the OpenPGP cryptographic suite) along with supporting libraries for our Web application, e.g. jQuery and AngularJS. The code we wrote to provide the webmail service, manage User Vaults, and provide redundancy, is closed source.

Open-source code provides a security advantage since a large number of people can access the source code and help to secure it by performing audits and reporting vulnerabilities. However, we believe the advantages of open-source code only apply to large projects with a strong supporting community. When a project is still too small to draw attention from external security experts, releasing the source code could pose a security risk that offsets the benefits. Potential attackers are given a powerful extra weapon: the source code of the application.

Therefore, we have chosen to keep our source code closed, as a security measure, and hire independent third-party auditors to verify our privacy and security measures. As StartMail grows, the potential benefits of opening up our source code may at some point outweigh the costs, and we will re-evaluate this decision at that time.

4. Security Measures

The previous section described the design choices that were made during the development of StartMail. In this section we explore the security measures that influence the organization, the development and maintenance processes, as well as the technical architecture.

4.1. Processes and Principles

We designed StartMail with privacy and security as our priorities. But no matter how secure a system is in theory, if a critical service contains a vulnerability, attackers can bypass carefully crafted security measures. With this possibility in mind, we instituted a security-oriented development process to ensure that vulnerabilities are not introduced while the code is being developed and maintained. We also added multiple layers of defense to limit the damage any one vulnerability can cause.

4.1.1. Development Process

Having secure code is a top priority in a project like StartMail. Of course, various measures exist to limit the impact that a vulnerability in our software could have.

We have multiple measures in place to make sure internal and external auditors can objectively assess StartMail's code:

- The programmers who work on StartMail have had training in secure development, and have experience in the field.
- High quality code that is well structured and easily readable is essential for security. Thus, we take great care to ensure that our code is clear for reviewers so they can understand the structure and flows. This allows them to focus on finding vulnerabilities.
- Each line of code is reviewed by multiple developers (other than the author) for quality and security before it enters the code base. Issues identified during this review cause the code to be rejected: It cannot move forward until the issues are resolved.
- Security professionals who are not part of the development team perform security audits of the code written by the developers. This is also the case for external libraries.
- The development infrastructure (code repository, issue tracker, review tooling) is used only for the StartMail project and is only accessible by its team members and auditors.

4.1.2. Open Source

We believe that using open-source libraries and tools, as opposed to ones controlled by commercial entities, is often the more secure choice. Solid communities are behind the development of these tools, and the available source allows us to internally audit them before considering whether to introduce them into our application. This is the main reason why, in addition to the code we write ourselves, we only use open-source components within the application.

Since the developers behind these tools also need to be supported in order to continue their development, StartMail has donated to open-source projects like LibreSSL [3].

4.1.3. Cryptography

Reinventing the wheel, when it comes to cryptography (and security in general), is a very bad idea [4]. It is not only unnecessary, but potentially harmful. History shows that cryptography is best left to standard libraries and tools, proven algorithms that have been subject to close scrutiny from the open-source and academic communities. Generally, developers are aware of this, but the issue is quite nuanced. Among the things we deem best left to cryptography experts, three categories are most notable:

1. Cryptographic algorithms for encryption, decryption and hashing. It is not a very good idea to try to replace AES, RSA or SHA-2.

2. Cryptographic protocols like PBKDF2, HMAC, TLS and various PKCS standards. Even when using proven cryptographic algorithms, it is usually unwise to come up with your own scheme to use them in.

3. Cryptographic implementations. Even when using textbook cryptographic algorithms, and remaining within the confines of proven protocols, the implementation is never trivial. We leave this to open-source libraries that are vetted by the community. In the case of OpenSSL, which received too little attention from the security community [5], we follow developments closely and will consider replacing OpenSSL as soon as a viable candidate becomes available.

4.2. Service Architecture

The previous section discussed security measures that take place at the organizational and process level. In this section we explain our consideration of the general architecture, webmail, account recovery methods, Business Accounts and IMAP.

4.2.1. General Architecture

4.2.1.1. TLS and PFS

In order to communicate with our users securely, we encrypt each connection using the TLS protocol. The exact details of how a connection to StartMail is encrypted depend on what the server and client negotiate during the setup of the TLS connection. By configuring secure preferences, the clients that support these requirements (which are all recent clients) will have secure connections with us. Our servers are configured with the following requirements:

- Require an important cryptographic property called Perfect Forward Secrecy (PFS). The importance of PFS is explained below. In practice, this means our servers prefer Diffie Hellman-based authentication ciphers like ECDHE.
- Only support hashing algorithms that have no known vulnerabilities that can be practically exploited, so no MD5.
- Prefer the use of the newest version of TLS: 1.2. This allows us to use stronger encryption ciphers (such as AES-GCM) and to protect against most attacks on block ciphers that have been published in the last two years.
- Prefer strong block ciphers (AES-GCM) over RC4. For backward compatibility with older clients, both are available, but when a recent client connects, AES-GCM is used.
- Use at least 128 bits of key material in block and stream ciphers (AES-GCM and RC4).

For a full evaluation of our TLS connection and practices please refer to our SSL Labs report [6].

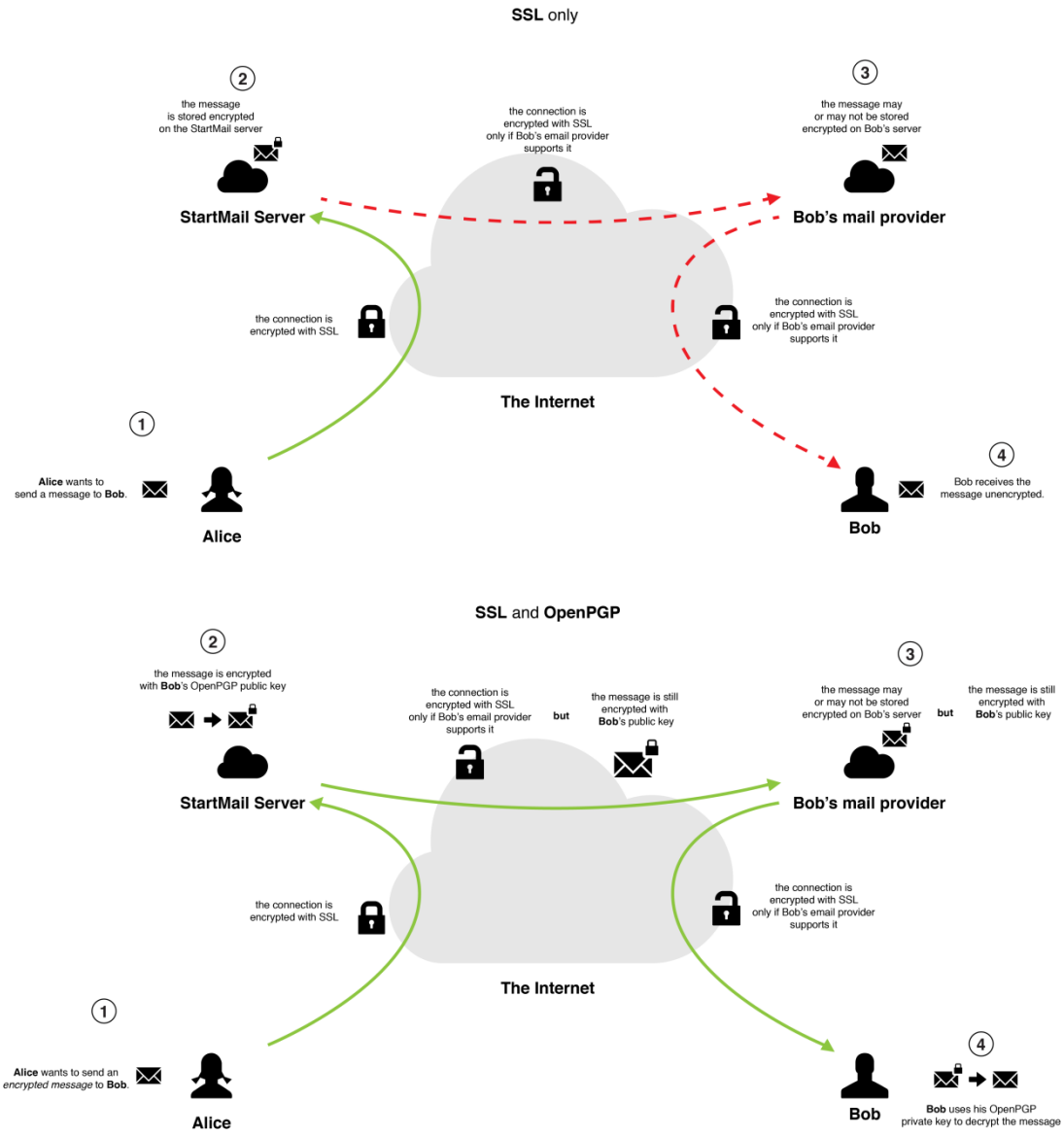
Whenever PFS is not used, there is a risk of previously encrypted traffic being compromised. That can happen if the private key of the system is compromised (whether by hacking, theft, or a government order), and the third party obtaining the key has recordings of previous traffic between clients and server. Perfect Forward Secrecy protects against this threat by preventing past traffic from being decrypted, even if the key is compromised.

StartMail uses the certificates from Buypass, a Norwegian Certificate Authority. A Certificate Authority based outside of the US gives us more certainty that StartMail certificates will not be compromised because of a government order.

4.2.1.2. Encryption in Transit and Storage

All connections to StartMail servers (and between StartMail servers) are protected by TLS encryption. Even though the connection is encrypted in transit, we recommend encrypting the emails themselves using OpenPGP so they can be stored encrypted while on the server or while being delivered via a less reputable external provider. Using OpenPGP is especially vital in communicating with others whose email providers are not committed to privacy and security, and when users access their email via IMAP (thus storing copies of their emails on a device).

The diagrams below illustrate the difference between encrypting the connection and encrypting the data itself, and why they are both important.



4.2.1.3. Header Stripping

As an added privacy feature we have implemented some header-stripping functionality for both incoming and outgoing mail.

For both incoming and outgoing mail we have the following measures in place:

- Obfuscation of local IP addresses and hostnames. This is done in order not to give out information about the infrastructure.
- Reject email coming from blacklisted "From" addresses. For instance, emails claiming to be from "administrator@startmail.com" or similar addresses.

Specifically for outgoing mail, we strip out the following headers:

- User-Agent
- X-Mailer
- Originating-IP

- X-Enigmail-Version
- X-Virus-Scanned
- X-Pgp-Agent
- Received
- Mime-Version

4.2.1.4. Rate Limiting

We have rate-limiting features in place to protect the system against password brute-forcing and username enumeration. A short-term (less than an hour) storage of IP addresses is used in order to identify potential attackers and limit the rate of password guessing. This is done in accordance with our privacy policy.

4.2.2. User Vault

User data is stored in what we call a User Vault, which is essentially a fully encrypted LUKS volume [7]. Users can access its contents for the length of their session by providing their account password.

While the Vault is closed (i.e. the volume is not mounted), no one can access the contents of the Vault, even if they otherwise have access to the server.

Because of our User Vault system, we do not have to store users' passwords, or even hashed versions of them, so we don't. Instead of checking a user's password, we simply use it to attempt to open the Vault. If this succeeds, the password was correct. Of course, the appropriate security measures are in place to protect ourselves against timing attacks when checking credentials this way.

4.2.2.1. What do we Store in the User Vaults?

We store the user's email messages and everything that is private about the user's account in the user's own User Vault (see our privacy policy for more details). This means that every time users want to access their Inbox, they must first open the account User Vault.

4.2.2.1.1 Mail

Email is stored in the User Vault when it is delivered to an account. Naturally, messages that arrive while users are logged out cannot be delivered until users manually open their User Vault by logging in.

Since we only want data to be accessible by the user it belongs to, an extra OpenPGP key pair is generated for every user and used as the "queue key pair". The private key is stored inside the User Vault, and the public key is accessible to the mailer service. When mail for a user arrives, it is encrypted with the queue's public key of that user and stored in a queue. Once the User Vault is opened (when the user logs in), emails are decrypted with the queue's private key and moved to the user's Inbox.

```
NOTE: The queue key pair is *completely separate* from the
account's OpenPGP key pair used for communications.
```

Of course, end-to-end protection is already in place when the sender of the message chooses to encrypt it using OpenPGP, which we always recommend. The process described above is simply an additional security layer.

4.2.2.1.2 Spam and Metadata

StartMail allows users to train SpamAssassin's Bayesian spam filter by marking emails as spam or not-spam in the webmail interface. The metadata related to spam filter training is still considered to be sensitive user data because it contains at least part of the contents of a user's emails. For this reason this data is stored inside the User Vault. This means that no one but the user can access this data, and that every user will have their own personally trained spam filter.

4.2.2.1.3 Keyring

If users opt in to use OpenPGP features in the StartMail webmail interface, their OpenPGP keyring will also be stored in their User Vault. They can add other (public and private) keys to their keyring for communication with other users.

4.2.3. Webmail Features

Through our webmail platform, we offer various features that enhance the security of messages. In this chapter we provide details about the implementation of these features.

4.2.3.1. Symmetric Encryption (Q&A messages)

Symmetrically encrypted messages can be created by composing an email and selecting Q&A as the encryption option.

Sending a Q&A Message is very similar to sending an encrypted email, with a few differences:

- The message will be encrypted using OpenPGP symmetric encryption, using the answer as a key.
- The encryption is done using GnuPG in symmetric mode, using the CAST-5 cipher, where the answer to the question is the passphrase supplied to GnuPG. CAST-5 is the default cipher used by GnuPG v1.0 & v2.0.
- Unlike a classic asymmetric OpenPGP encrypted message that needs to be encrypted for all the public keys of the recipients, this encryption option is available for any recipient - even those for whom no OpenPGP public key has been imported.
- When the designated recipient of the message is not a StartMail user, the message is not actually sent out as an email. Instead:
    - It is stored encrypted on our servers.
    - A notification email is sent out to the recipient of the message. This notification contains a link to a page on a StartMail server.
    - When the recipient clicks the link in the notification email, the sender's challenge Question appears with a prompt requesting the Answer. The Answer must match the Answer specified by the sender.
    - The recipient has only five (5) opportunities available to provide the correct Answer, after which the message will no longer be accessible. This is done to prevent brute-forcing.
    - Once the recipient provides the correct Answer to the Question, he or she will be able to view the content of the Q&A message and reply securely through the same interface.
    - If the sender has enabled OpenPGP functionality in StartMail, the (optional) reply is asymmetrically encrypted with their public key.

If the designated recipient of the message is a StartMail user, the message will be sent internally as a normal encrypted message. The OpenPGP functionality embedded in StartMail will allow the recipient to decrypt it in the Web interface by providing the Answer to the Question.

As an additional security measure, Q&A messages are automatically deleted from the StartMail server after a set amount of time. Users can select the time frame for expiration ranging from a day to three months. This expiration only applies to Q&A messages sent out to non-StartMail users. Since we can safely store the messages for StartMail users inside their Vault, we do not need to (nor have the ability to) delete the messages after the configured time.

4.2.3.2. OpenPGP

We offer OpenPGP functionality [8] in our webmail interface. OpenPGP is a standard protocol, described in depth in RFC 4880 [9]. According to the OpenPGP standard, two separate keys are used for encrypting and decrypting data. Every user has their own key pair: One private key which is used, along with a secret passphrase, to decrypt messages and files; and one public key, designed to be given to other people. Messages encrypted with a particular public key can only be decrypted with the associated private key.

Once users activate OpenPGP functionalities, they are prompted to either generate a new key pair or import an existing PGP key pair.

4.2.3.2.1 New OpenPGP Key

We use the standard GnuPG tool to generate a new key pair (a public and private key) on the server. The key pair's associated email address is, of course, the user's StartMail address.

The keys that are generated by StartMail are 4096 bit RSA keys.

4.2.3.2.2 Imported OpenPGP Key

Users who already have an OpenPGP key pair can import it into StartMail. The StartMail email address needs to be associated with the key pair before importing it.

4.2.3.2.3 OpenPGP Key Management

Basic key management is available in the StartMail Web application:

- Changing the private key's passphrase
- Importing/exporting keys
- Regenerating keys

4.2.3.2.4 Passphrase Caching

The private OpenPGP key is stored encrypted on the server inside the User Vault. Additionally, the key itself is always encrypted with a passphrase. The use of OpenPGP in the Web interface means that the private key's passphrase will have to be available to the application so that decryption and encryption operations can take place on the server. Whenever the passphrase is needed, users are prompted to type it in and can choose to optionally keep it in memory. The available options for caching the passphrase are:

- Never
- 5 Minutes

- 30 minutes
- 120 minutes
- Until logout

If a caching option is chosen, the passphrase is encrypted and stored in-memory on the server. The encryption key for the passphrase is stored in a cookie in the browser. This means that an attacker who gains access to a user's cookie still cannot recover the passphrase itself.

Obviously, not caching the passphrase at all is the most secure option, if somewhat inconvenient, but we leave this choice to the user.

4.2.3.2.5 OpenPGP Key Directory

When users let StartMail handle their OpenPGP operations, the public key is automatically added to an internal key directory. When the user composes an encrypted email message to another StartMail user who has also set up OpenPGP in StartMail, the public key of the recipient is retrieved from the key directory. Users can opt out from this key directory in their preferences.

```
NOTE: The public keys are not shared with StartMail users, nor
are they publicly available. They are only used to transparently
encrypt communications *among* StartMail users. However, users
are additionally given the option to upload their OpenPGP public
key to the MIT OpenPGP Key Server [10].
```

4.2.3.2.6 Signing and Verifying a Signature

Along with email encryption and decryption in the StartMail webmail interface, we provide the possibility of signing emails with OpenPGP signatures. Signing emails is very important: It provides a way to verify the sender's identity by guaranteeing that the person sending the encrypted email is the owner of a private key that matches the sender's email address. By default, all OpenPGP encrypted emails sent via the StartMail webmail interface are also signed.

The StartMail webmail interface provides visual feedback about the validity of signatures on email messages and warns users when they receive messages with an invalid signature. In order to verify signatures of other StartMail users, their OpenPGP public key is automatically retrieved from the internal key directory mentioned in the previous section.

4.2.3.2.7 OpenPGP Keyring in Vault

As previously mentioned, one of the items stored in the User Vault is the OpenPGP keyring. This is important because without opening the User Vault, the user's own key pair is not available to the system. The keyring contains the user's own key pair and all other keys he has uploaded or imported.

4.2.3.2.8 OpenPGP and BCC Receivers

With traditional PGP encrypted email, using the BCC field to declare recipients that should not be exposed to others is impossible. This follows from the fact that an encrypted message also includes the key IDs for which the message has been encrypted. Any recipient with knowledge about the key IDs can also determine who was included in the message by inspecting the keys listed in the encrypted message.

There are several ways to prevent this leaking of data. We chose to extract all BCC receivers from a message sent through our webmail service. We then create and encrypt a separate email for each individual BCC recipient and leave them out completely from the original message that is sent. This way, no information about the BCC recipients will leak to the other primary recipients.

4.2.3.2.9 PGP Message Type (MIME/Inline)

We fully adopt the PGP/MIME message structure for all outgoing messages because it cleanly encrypts all attachments without exposing metadata about them. Furthermore, it is more reliable for non-ASCII text and works properly with HTML email.

Finally, PGP/MIME is generally considered the most appropriate method and allows all mail clients to properly display the message, whether they have PGP support or not.

If we receive a message encrypted or signed with PGP/Inline, we can correctly decrypt it (given we have access to a matching private key) and verify signatures (given we have access to the public key). When replying or forwarding PGP/Inline messages, we convert them to PGP/MIME.

Although PGP/MIME is known to be unsupported by some older email clients, we value the privacy and security improvements of this structure and consider them to be more important than compatibility with clients that have limited MIME compatibility.

4.2.3.3. Virus Scanning

On most mail services, encrypted email is not scanned for viruses. StartMail, in contrast, scans encrypted messages for viruses once they are decrypted using StartMail's Web interface. Messages that are found to contain a virus are kept in the user's inbox, but they become inaccessible (though they can still be downloaded via IMAP), and the user is alerted to the threat.

4.2.4. Recovery

Recovery options differ for Business Accounts and Personal Accounts. For Personal Accounts we have recovery options for the account password that the user chooses. For Business Accounts we also provide a method for recovering PGP passphrases for the sub accounts (Business Account mailboxes), which we explain later in the Business Account section.

4.2.4.1. Personal Account Password Recovery

Should users ever forget their account password, we offer the possibility to recover it:

- Using a recovery code
- Having a reset link emailed to their recovery email address

During signup, users may choose between either obtaining a recovery code or setting a recovery email address. The recovery code is provided if the user chooses not to set and confirm a recovery email address.

It is important to note that regardless of the user's choice, a recovery code will be generated automatically by the system. The user will be completely responsible for its safekeeping. If the user does not set and confirm a recovery email address, StartMail will not store the code.

When the user designates and confirms a recovery email address, StartMail will store the code encrypted. We explain the process in detail in this section.

4.2.4.1.1 Recovery with a Recovery Code

The recovery code, obtained during signup, can be used to manually reset the account if the user forgets their account's password.

StartMail does not store recovery codes for accounts that do not set and confirm a recovery email address. Therefore, our staff cannot help users who lose their recovery code. It is technically impossible for us to manually reset accounts, and, even if it were possible, we would have no way to verify the identity of the person requesting the recovery.

Whenever giving out recovery codes this way, the responsibility of secure storage shifts to the users. We recommend that users who feel they may have trouble safely storing the recovery code themselves, provide a recovery address and allow StartMail to handle the storage and the recovery for them, should it be necessary.

4.2.4.1.2 Recovery with a Recovery Address

Address Verification

After signup, if a recovery address is added, a verification email is sent out to the recovery address. Confirming that recovery email address is very important because it allows StartMail to verify the ownership of the email address set for recovery. Once the address is verified, users can proceed to request a recovery, should it become necessary.

```
NOTE: Users must confirm their recovery email address before
attempting to recover their account, or we will not be able to
proceed with the recovery process.
```

Recovery Code Storage

The recovery code is generated automatically by the server and stored in an OpenPGP encrypted file, using multiple layers of encryption. (The next section explains why this is relevant.) No one, including StartMail support staff or administrators, can access the unencrypted recovery code, nor can anyone manually produce a recovery code associated with any account.

We deem it very important to shield the recovery code from human access. The recovery code effectively gives whoever possesses it the power to reset an account's password. An attacker in possession of such a code could take over the account completely. We take every precaution necessary to make sure it can only be obtained by the legitimate user.

More Than One Person Required to Decrypt

The process of authorizing a recovery request is not entirely automated. It requires two separate senior members of the management team to decrypt (remove a layer of encryption) and acknowledge the request. The people involved reside on separate continents (EU and US) to keep them under different legal jurisdictions.

This is done to prevent a single person from having the power to approve a request. Should the authority of one person be compromised, additional interaction is always required in order to proceed with the account recovery.

Once the second required manager has acknowledged the recovery request and decrypted the recovery file with their key, the resulting file is forwarded to the StartMail system. The StartMail system then removes the last layer of encryption and sends out the recovery code to the verified email recovery address for the account.

4.2.4.2. Recovering the Account

The account recovery process involves changing the password of a user's User Vault. The Vault is a LUKS volume [7], which has several key slots: one slot is used for authentication, another for recovery. A successful recovery will result in the following:

- The Vault is unlocked using the key corresponding to the recovery slot
- The old authentication slot is overwritten by setting a new password
- The old recovery slot is overwritten by generating a new recovery code

4.2.5. Business Accounts

Business Accounts are for businesses and individuals who own a domain and would like to use StartMail for managing their domain-based email. A Business Account also comes with some special features that we detail in this chapter.

4.2.5.1. Recovery

With traditional PGP, loss of a passphrase means losing access to encrypted email, which is a potentially catastrophic event for a business. StartMail has engineered a way for Business Account Administrators to recover PGP passphrases, as well as account passwords. For security reasons, StartMail team members are prevented from resetting passphrases and passwords, and cannot do so even by court order.

4.2.5.1.1 Master Key Generation

During account set up, the Business Account Administrator receives a “Master Key”. This Master Key is a randomly generated passphrase for a special key pair (RSA) that also protects all the passwords and passphrases related to a Business Account. We chose to generate this Master Key to prevent users from choosing one that would be easily guessable because it would be extremely unwise given the enormous power of this key. The assigned Business Account Administrator should be a highly trusted individual for the same reason.

The Master Key is shown to the Business Account Administrator once during initial setup of the Business Account and needs to be stored securely because someone using the Master Key will be treated as a trusted Business Account Administrator by the StartMail system. The Master Key cannot be recovered by the business user or by the StartMail team.

4.2.5.1.2 Recovering Account Passwords

Every time a business user creates a new Staff Account, the StartMail system automatically generates an account recovery code. This recovery code is then encrypted with a random key, which is then encrypted using the Master Key. This encrypted recovery code is stored in a secure StartMail database.

When a Staff Account password is lost, the Business Account Administrator can recover the account by supplying the Master Key and a new password. If the Master Key is correct, the primary LUKS passphrase slot is overwritten with the new password. The Business Account Administrator can then provide the new password to the Staff Account user.

Business Account Administrators can also recover their own account passwords following these account recovery steps. Recovery can be performed from within the Administrator interface (when logged in), and through the StartMail recovery page (when not logged in.) The former is meant to be user friendly; the latter will allow a locked out Administrator to regain access.

4.2.5.1.3 Recovering PGP Keys

Our goal is to empower businesses to use PGP without having to worry about the traditional usability issues. Because one of the most common issues with PGP is a user losing his PGP passphrase, we have implemented a specific solution for Business Accounts.

Again, we use the Master Key, this time as the means to encrypt the PGP keys of all Staff Accounts. Whenever a Staff Account user creates or imports a PGP key pair, StartMail automatically stores the PGP key pair in the database, encrypted with the Master Key, using the Staff Account passwords methods described earlier.

Business Account Administrators can recover Staff Account PGP key pairs through the StartMail Administrator interface by supplying the Master Key and entering a “temporary recovery” PGP passphrase. If the Master Key is correct, StartMail automatically emails the recovered key pair to the Staff Account user. The Staff Account user can then easily import the key pair from the email, and then set a new PGP passphrase.

```
We deemed using email for this PGP passphrase recovery
appropriate because the PGP key pair is transferred directly into
the user’s inbox and is stored encrypted in either the User’s
Vault or in the incoming queue. Additionally, the key pair is
further protected by the Master Key.
```

4.2.5.1.4 Privacy and Security Considerations

StartMail’s unique PGP passphrase recovery system for Business Accounts treats PGP keys as private as they relate to the business, rather than the end user. We chose this approach because business information generally belongs to the business and should always be accessible to the business. This approach also supports the sharing of accounts and PGP keys in the office setting. We recommend alerting employees to this so they don’t inadvertently use their accounts to communicate personal information they would be uncomfortable sharing.

Businesses that are uncomfortable treating PGP keys as business property can use PGP completely outside the realm of StartMail by using IMAP and client-side PGP.

4.2.5.2. Special Email Addresses

All domains that handle email are required to at least receive email by default on the abuse@… and the postmaster@… addresses. Both are required mostly for technical reasons. For every domain that is linked (and verified) in a Business Account, we set up both addresses as default aliases for the Administrator Account(s).

4.2.6. IMAP and Mobile Devices

Users who wish to access their email through a separate email client can do so through IMAP. IMAP access is disabled by default, and can be enabled in the Settings menu.

```
NOTE: Many users accessing their email from a separate email
client are used to the POP3 protocol. IMAP works in a very
similar way, but is designed to keep messages stored on the
remote server until they are deleted, as opposed to always
downloading them to the client machine and deleting them from the
server.
```

IMAP (as opposed to POP3) integrates more naturally with viewing email in StartMail's Web interface and allows users to still benefit from the secure storage that the User Vault system offers.

We recommend configuring every external device separately that connects to StartMail via IMAP. Once users add a (named) device, they will obtain a unique username, password and IMAP server information combination that they can use to configure the device.

Creating separate IMAP credentials for every device has several advantages in case one of the IMAP-enabled devices (laptop, phone) is compromised:

- An attacker in possession of valid IMAP credentials only has access to a user's messages. There is no risk that the entire StartMail account can be taken over by an attacker in this scenario.
- IMAP accounts can be independently disabled.

Creating a different password for each device that connects to StartMail also offers an overview of which external services are accessing StartMail. Should a service ever become lost, obsolete, or unused it is simple to remove the device and revoke access.

4.2.6.1. Automatic Disabling of a Device

After numerous unsuccessful authentication attempts, a device will be disabled. Users can verify that a device has been disabled by logging into their StartMail account, going to the Mobile/IMAP Settings page, and verifying that the device in question is marked as disabled.

Once a device has been disabled, it cannot be re-enabled. A disabled device can be set up again as a "new" device.

4.2.7. Infrastructure Security

Our infrastructure is strictly based in the Netherlands and has been built to support the security requirements of the StartMail service.

As a general rule, we isolate services and components of StartMail as much as possible in order to contain the potential damage an attacker can do. For instance, User Vaults are stored on separate servers, in terms of infrastructure, from the Web servers. They communicate with each other through a very limited API to minimize the damage that can be done by someone compromising a Web server.

Furthermore, we have taken all standard precautions that would be expected in a secure hosting environment, such as internal (PFS) TLS communications, strict firewalls, full disk encryption on all machines, and so forth. We have also taken extra measures to anonymize logs, as described in detail in our privacy policy.

Finally, we use active logging and alerting, integrated with a kernel-level audit system that alerts us to anomalous activities on all the servers. These logs are regularly reviewed for activity that might indicate a compromise. In addition, failing to acknowledge audit log messages in a timely manner results in a monitoring alert itself.

5. Conclusion

The people behind StartMail have many years of experience in offering privacy-enhancing services. Our goal was to create an email service that truly embraces the promise of "encryption made easy," to help people and their businesses regain their right to secure and private communications. Creating a platform to make encrypted email communications available to the average user was a process of finding the optimal combination of user friendliness, privacy and security.

We use proven cryptography libraries and rely on standard implementation to provide the underlying security structure. Our development process is entirely focused on eliminating vulnerabilities before they enter our codebase. Our very strict (and complex) account recovery process prevents unauthorized users from having the ability to reset accounts. One of StartMail's most important features is the user's personal User Vault, which allows us to encrypt all the data that belongs to the user.

We made carefully thought out choices, such as selecting server-side cryptography versus browser-based JavaScript solutions. We have thoughtfully considered where to store OpenPGP private keys to ensure maximum security for the non-tech-savvy users. We have weighed enhancing our search and spam-detection algorithms against protecting email contents, and we have considered whether to release StartMail's source code or remain closed source. In every trade-off, when given a choice, we have made -- and always will make -- the decision that favors the security and privacy of our users.

Our approach to introducing new technologies into our system is a very conservative one. We only rely on proven solutions that the cryptographic community has confidence in to securely protect our users' privacy. At the same time, we keep a keen eye out for new opportunities to enhance our security practices. We periodically re-evaluate our decisions in the light of the new technologies as they become available so we can continue to offer the most secure, state-of-the-art service possible.

6. References

1: https://www.startmail.com/privacy "StartMail Privacy Policy"

2: https://en.wikipedia.org/wiki/Fault_tolerance “Fault tolerance” (Graceful degradation is a particular type of implementing fault tolerance)

3: http://www.openbsdfoundation.org/contributors.html "OpenBSD contributors list"

4: http://www.openbsdfoundation.org/contributors.html "Why shouldn't we roll our own?"

5: http://www.openbsd.org/papers/bsdcan14-libressl/ "LibreSSL - The first 30 days"

6: https://www.ssllabs.com/ssltest/analyze.html?d=startmail.com "StartMail Qualys SSL Labs report"

7: https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup "Linux Unified Key Setup"

8: http://en.wikipedia.org/wiki/Pretty_Good_Privacy "PGP Encryption"

9: http://tools.ietf.org/html/rfc4880 "RFC 4880"

10: https://pgp.mit.edu/ "MIT PGP Public Key Server"