University of ???

PhD Thesis

Possible solutions to implement email transfer offering anonymity towards third parties

Author:
Martin Gwerder

Supervisor:
unknown

January 27, 2014

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This document describes a solution, which should offer anonymity against third parties when sending emails based on SMTP and the respective client protocols (e.g. IMAPv4 or POP3). This seemed to bother very few peoples up until information in Echelon became public due to an investigation by a commitee of the european parlament in 2001[5]. Things settled again with local peaks up until a whistle blower named Edward Snowden disclosed 200000 documents proofing activities of the NSA and other secret services. This led to the "2013 mass surveillance disclosures" and damaged the reputation of the american nation in many countries[34].

1.1 Overview over the current situation

SMTP as defined in RFC5321[17] is as of today (2013) state of the art transmission protocol for electronic mail. It is standardized in its current version since 2008 and is one of the few protocols, which is marked as "Standard". While the protocol delivers reliable mail transfer between two endpoint (mail servers) the anonymity of the message content towards any mail server is not given (For a detailed analysis see 4).

Anonymity against third party is not given due to the following facts.

- There is not always an encryption available between a mail user agent (MUA) and the outgoing mail server.
- There is no way to guarantee that a mail transfer between two SMTP hosts is encrypted.
- There is no always an encryption available between a SMTP host and the MUA of the recipient.

- Encryption based on top level protocols (such as S/MIME or PGP) do hide the message content. The sender, recipient, the subject and some technical information (eg. MIME-Headers) are always in plain available and not protected as such.
- Even if there is a reliable encryption between all endpoints and none of the intermediate servers are compromised sender and recipients might still be identified thru traffic analysis.

Keeping the message content confidential is more and more relevant in these days. The more the importance of mail transfer in today's economy is growing the more is confidentiality and reliability a topic. Unfortunately Secret Services have already discovered the significance of today's mail traffic and start to analyse those. With the presence of Secret Services in the internet, actively investigating data the importance of a reliable data channel for today's messages has become increasingly important.

Quick wins such as the use of "Onion Router Networks" (such as TOR) do not offer any additional security since the message content would be revealed in full to an eventual exit node and any mail server on its way to the recipient.

1.2 Problem statement

This work is an approach to extend the existing mail routing based on SMTP by an intermediate layer, which should offer anonymity against third party.

This work delivers the following results:

- A throughout analysis of current technology and its weaknesses.

  Although the Simple Mail Transfer Protocol (SMTP) is a well-implemented and well proven technology its weaknesses are well known. The SMTP protocol was originally defined in RFC821[28] by Johnathan B. Postel. At this time internet was only available to universities, some mayor companies and governments. The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently[28, p. 1]. Confidentiality or having a tamper proof protocol was no design goal. Over the years many standards arose trying to close some of the gaps. Some of them are being used but most of them are not very common.
- An analysis of possible approaches to improve the current standards.

  Many standards and technologies do exist these days addressing parts of the issues mentioned above. A throughout research should be carried out to identify how can these technologies be combined to achieve the subsequent goals. Furthermore technology advanced. Namely in the field of cryptology few possibilities and ideas arose (such as new encryption classes [eg. elliptic curves] or the idea of crypto puzzles). Another field of research which emerged in the analysis of traffic flow is handled under the term "Big Data" where not single events but the sum of events is handled.

- A RFC document

  It will describe an approach offering a significant quality improvement of the existing solutions, which could be accepted by the internet community.

  The document has to follow the standards FIXME

- A prototype reflecting at least the minimum baseline of the RFC document to reflect prove its functionality.

  A prototype should be offered to show the feasibility. The Prototype should be a reference implementation and offer a quick way to use the new technology. It should be distributed under the LGPL license to simplify distribution of the technology.

1.3 Contributions

This thesis contributes to the topic in the following senses:

- It introduces a consistent model for message delivery which includes all endpoints

- It shows an approach based on existing protocols for anonymous communication

1.4 Notation

The theory in this document is heavily based on encryption and hashing. In order to use a uniformed notation I use $E^{K_a}(M)$ for a symetric encryption of a message M with a key K_a where a is an index to distinguish different keys. Decryption uses therefore $D^{K_a}(M)$.

As notation for asymetric encryption I use $E^{K_a^1}(M)$ where as K_a^{-1} is the private key and K_a^1 is the public key. The asymetric decryption is noted as $D^{K_a^{-1}}(M)$.

For hashing I do use $H(M)$ if unsalted and H^{S_a} if using a salted hash with salt S_a. The generated hash is shown as H_M if unsalted and $\{S_a, H_M\}$ if salted.

$$D^{K_a^1}\left(E^{K_a^{-1}}(M)\right) = D^{K_a^{-1}}\left(E^{K_a^1}(M)\right) = M \tag{1.1}$$

$$D^{K_a}\left(E^{K_a}(M)\right) = M \tag{1.2}$$

$$H(M) = H_M \tag{1.3}$$

$$H^{S_a}(M) = \{S_a, H_M\} \tag{1.4}$$

Chapter 2

Ground theory

2.1 Mail Transport

Todays mail transport is mostly done via SMTP protocol as specified in [17]. This protocol has proven to be stable and reliable. Most of the messages are passed from a MUA to a SMTP relay of a provider. From there the message is directly sent to the SMTP server of the recipient and from there to a server based storage of the recipient. The recipient may at any time connect to his server based storage and may optionally relocate the message to a client based (local) storage. The delivery from the server storage to the MUA of the recipient may happen by message polling or by message push (where as the later is usually implemented by a push-pull mechanism).

To understand the routing of a mail it is essential to understand the whole chain staring from a user(-agent) until arriving at the target user (and being read!). To simplify this I used a consistent model which includes all components (server and clients). The figure 2.1 shows all involved parties of a typical Mail routing. It is important to understand that Mail routing remains the same regardless of the used client. However – Availability of a mail at its destination changes drastically depending on the type of client used. Furthermore control of the mail flow and control is different depending on the client.

In the following paragraphs (for definitions) the term "‘Mail"’ is used synonymously to the term "‘Message"’. The reason why "‘Mail"’ has been chosen over "‘Messages"’ that a lot of terms do already exist in standard documents. In these documents the term mail is commonly used.

Mails are typically initiated by a Mail User Agent (MUA). A MUA accesses a local mail storage which may be the server storage or a local copy. The local copy may be a cache only copy, the only existing storage (when mails are fetched and deleted from the server after retrieval) or a collected representation

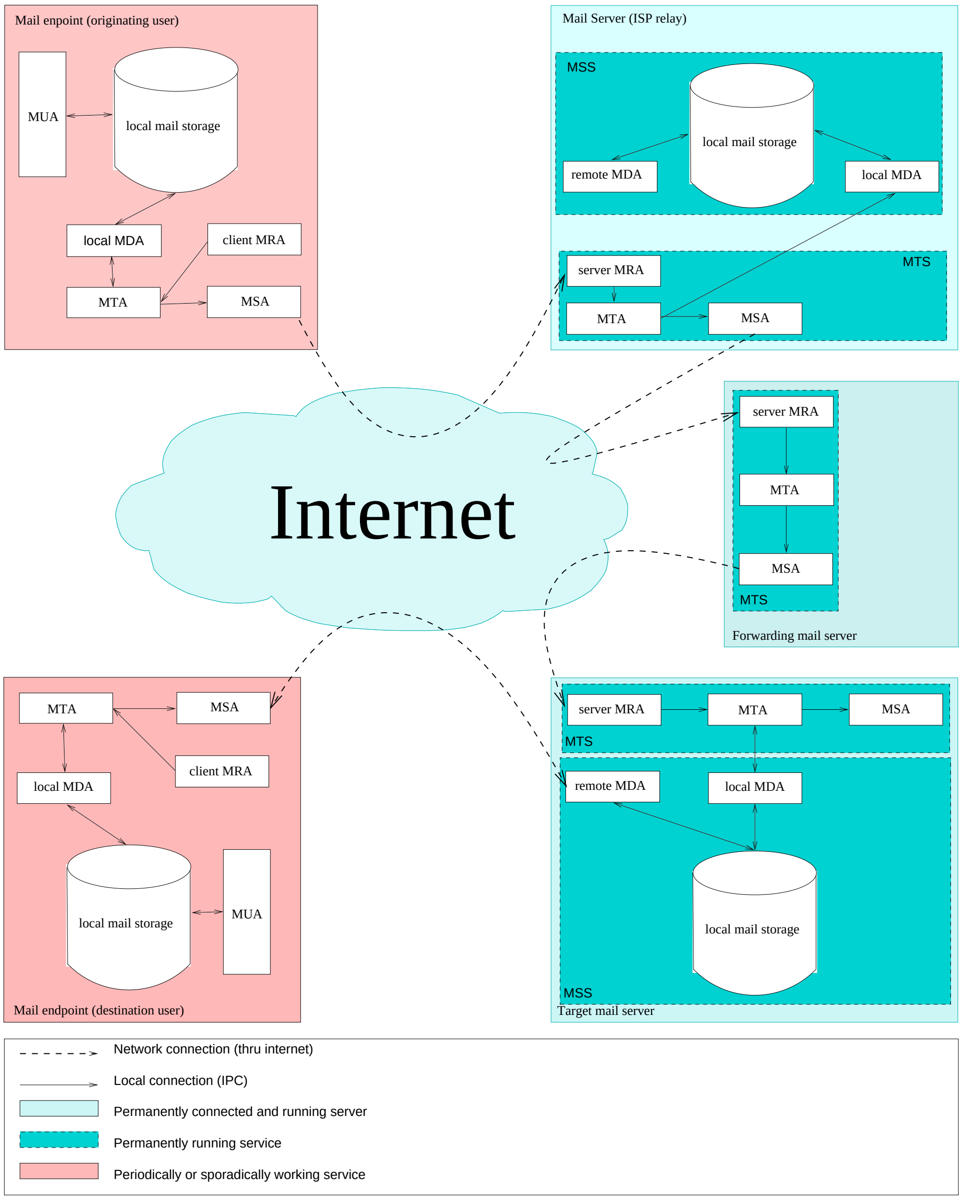

Figure 2.1: Mail Agents

of multiple server storages (cache or authoritative).

Besides the MUA the only other component accessing a local mail storage is the Mail Delivery Agent (MDA). An MDA is responsible for storing and fetching mails from the local mail storage. Mails destined for other accounts than the current one are forwarded to the MTA. In the case of a rich client the local MDA is part of the software provided by the user agent. In the case of a mail server the local MDA is part of the local Mailstore (not necessarily of the mail transport service).

On the server side there are usually two components (software sets) at work. A “Mail Transport Service” (MTS) responsible for mail transfers and a "‘Mail Storage System"’ which offers the possibility to store received Mails in a local, persistent store.

A MTS consists generally out of three parts. An server Message Receiving Agent (Server MRA) is typically a SMTP listening daemon. A Mail Transfer Agent (MTA) which is responsible for routing, forwarding and rewriting mails. And a Mail Sending Agent (MSA) which is responsible for transmitting mails to another Server MRA (usually done by SMTP).

A MSS consists out of a local storage and delivery agents which do offer uniform interfaces to access the local store.

2.1.1 Mail clients (MUA)

2.1.1.1 Fat clients

2.1.1.2 Server located clients

2.1.1.3 Web clients

2.2 Anonymity

2.2.1 *k-anonymity*

2.2.2 Plausible deniability

2.2.2.1 Deniable encryption

2.3 Identification and data signage

2.4 Encryption

2.4.1 Key exchange

2.4.1.1 Diffie-Hellmann key exchange

2.4.2 Symetric encryption

2.4.2.1 Advanced Encryption Standard

2.4.3 Asymetric encryption

2.4.3.1 RSA

2.4.3.2 El-Gamal

2.4.3.3 ECDSA

2.5 Mix cascades

2.6 Remailers

Agents which do accept Mails from one party and forward it to another party while modifying its content well known under the name of "Remailers". Wikipedia [33] lists four types of Remailers.

Pseudonymous Remailers (or Type-0-Remailers) are remailers that establish a pseudonymity. This means that the senders Email-Address is removed and replaced by a pseudonymous E-Mailadress under the remailers control. This sender address may be used as an ordinary email-Adress to reeach the original sender of the mail. These types of Remailers allow to send mails while one or both recipients do not know their counterpart. The message (or at least parts of it) might be encrypted but do not have to be. For someone controling the Remailer it will always be possible to make a link between the pseudonymous mail address and a original mailadress. So pseudonymity is only granted towards people not controlling the remailer. Furthermore a person or organisation might be able to discover the Information tuple of Sender and pseudonymous email by analyzing messages and their timely context. So this remailer system is suspectible for traffic analysis.

Cypherpunk-Remailers (or Type-1-Remailers) do function a bit different. They take an encrypted message which was encrypted using the public key of the server, decrypt it and send it to a recipient. The original senders identity gets lost. A reply to a cypherpunk message is not possible. Messages sent to a cypherpunk server might contain messages to other cypherpunk remailers. This daisy-chaining of cypherpunk-nodes allows hiding the original sender-receiver-tuple from a single node. The first node knows only the the originating sender while the last node knows only the final recipient. All intermediate notes do only know the nodes they were linking. However if having traffic information of the entry and exit nodes the tuple might be discovered by traffic analysis.

Mixmaster remailer (or type-2-remailer) is a serie of mailers which split up a message into equally sized chunks and forward them using different paths (via SMTP) to an exit node where the message is reassembled and sent to the final recipient. However if having traffic information of the entry and exit nodes the tuple might be discovered by traffic analysis.

Mixminion remailer (or type-3-remailer) is an enhanced development of Mixmaster remailer. It is currently no longer under active development. It adresses serveral weaknesses of the mixmaster. Namely replies are possible. Forward anonymity is now given. Replay prevention and key rotation is part of the design and there are exit policies allowing ISPs to opt out from receiving remailer traffic. It is based on a proprietary communication network. It furthermore introduces dummy traffic to reduce traceability. This is the most complete approach email anonymity ever given. The aproach has however its weaknesses. To avoid partitioning attacks Miximinon distributes its network information with central redundant directory servers.

2.7 Ethics

2.7.1 Human rights

2.7.1.1 Freedom of speech

Article 19 of the ICCPR states that "everyone shall have the right to hold opinions without interference" and "everyone shall have the right to freedom of expression; this right shall include freedom to seek, receive and impart information and ideas of all kinds, regardless of frontiers, either orally, in writing or in print, in the form of art, or through any other media of his choice".

2.7.2 Ethics of the Internet

There is an RFC document regarding “Ethics and the Internet”[31, p. 1]. Document states as unethical behaveour:

- An activity that seeks to gain unauthorized access to the resources of the Internet.
- An activity that disrupts the intended use of the Internet.
- An activity that wastes resources (people, capacity, computer) through such actions.
- An activity that destroys the integrity of computer-based information.
- An activity that compromises the privacy of users.

Unfortunately these actions do exist in modern internet and the most powerful players discovered so far are governmental agencies. Using a mixer and cryptographic algorithms definitely wastes resources. But it must be considered the right of every single user of the internet to uphold these points. As a final conclusion the proposed system does not violate the ethics of the internet but it must be designed to be as economically as possible with the existing resources.

2.8 Possible legal issues

One of the first questions I have been asked when working for this topic was: Is this legal? The question is important but not easy at all. The mail system is a global spanning network coming across almost any country of the world. Some of these countries consider almost any kind of secret as illegal as long as the country itself is not able to capture it. Some countries consider it as perfectly legal and some will generally accept its presence as long as the country or establishment is not endangered due to its usage. Since there is usually control about mail traffic flow there is no mean to tell what laws have been violated by sending a mail. This is not specific to this work but a general problem which ocures often in connection to the internet.

FIXME <should be removed>

My personal unscientific point is: I do not care. In my country it is definitely legal as long as I am a well behaved citizen (as long as I do not missuse this system to plan or do illegal actions). There are already proprietary systems available which offer the same functionality. All I do is adding this functionality to the common system instead of reinventing the wheel. There are however many very good reasons to have such a system. Correspondence about my health, my business relations, my friends or my family (to give just a couple of examples) should be kept private even in an open world. The misuse of information would cause tremendous damage and several events in time (which have been mentioned earlier) showed that there are many secret services and other players using any kind of information to achieve their own goals or the goals of associates. They do this regardless of any country borders or regulations. Since I have no means of controlling the flow of messages in the internet or the hubs where a mail is running thru I consider it as fair to generate an addon to compensate the lack of control in the existing system. Exactly as a car – the system may be legal or illegal and it depends on the users whether he wants to use it or not and in what way.

FIXME </should be removed>

Chapter 3

Current situation

As of today the de facto standard for asynchronous mail transfer is SMTP as defined in RFC5321[17] and its predecessors. While the transfer protocol SMTP is quite compact, the protocol is enhanced with several standards for encryption, multimedia support and similar. A mail client offers today various support for a lot of sub-protocols. The following list is an excerpt of related sub-protocols which are either related to transport, reliability, identification or encryption.

3.1 Implemented protocols

3.1.1 SMTP

The SMTP protocol is currently specified in [17]. It specifies a method to deliver reliably asynchronous messages thru a specific transport medium (most of the time the internet). The Protocol makes a dstinction between a mail envelope and its content. The envelope contains the routing information which is the sender and the recipient. The content again is split into two parts. These parts are the headers (which do contain meta information about the message such as subject, reply address or a comprehensive list of all recipients) and the body which contains the message itself.

It furthermore introduces a simplistic model for mail communication. A more comprehensive model is introduced in the section Mail Transport. As the proposed model is not sufficient for a comprehensive end-to-end analysis.

FIXME incomplete section

3.1.1.1 Mail transport

[18]

3.1.1.2 encryption

Encryption is the only anonymizing technology which is available. There are serveral kind of encryptions which have to be differentiated. Link encryption controls the E-Mail connection a guarantees that the whole communication between two servers is encrypted. It does however not guarantee that the message and routing information is protected all the way thru the network. Message encryption is a weaker encryption which is done at a higher level of the protocol stack. It guarantees that a message is end to end encrypted but discloses all routing and header information.

One kind of Mail link encryption is specified in [15]. This RFC specifies that when a STARTTLS-Command is issued a TLS handshake initiating a encrypted link should be carried out between two Servers. Only not public servers (not published in DNS using MX records) may enforce the use of TLS. All public servers must allow non-TLS transport. Authentication thru this port is possible but usually not done. The STARTTLS specification states clearly that securing a link provides no end-to-end security. An attack to this mechanism is very simple. The only thing required is injecting a 454 error code when the client issues a STARTTLS. According to the document the sending server may then refuse to deliver the document but in reality this never happens in public SMTP servers.

For encryption between a mail endpoint (repective its MSA) and the server MRA Clients may choose to use alternate ports which enforce a TLS handshake at the TCP handshake. This invalidates the possibility to disturb a connection while still in plain text modes with fake errorcodes but since it is a weak security anyway it makes really a difference. According to the [15] document the port 587 should be used. On some servers the same functionality is provided on port 465. This was originally intended for mail transmission between two MSAs. The usage of this port has however never been standardized, violates [14] and the port has been assigned to the URD Protocol by IANA.

The second type of encryption is message encryption. Message encryption does not cover the whole server communication starting from a specific point. It does only cover some parts or the full message body. The Two main protocols in use are S/MIME (As specified in [3]) and PGP/MIME (as specified in [6]; bases on [13]). Both do reveal vital information to all involved parties of the mail transport and a possible third party observer thus completely invalidating anonymity. Informations which are visible to anyone are:

- sender address (may be forged)
- sender client (may be forged)
- Recipient address
- message subject
- the full routing path including all rewrites, timing information and intermediate hops.
- the content type
- Mime-Version
- Date and time of sending

Any client or intermediate Server may furthermore add additional information of any kind (such as virus scanning information, anti spam taxation, reply address).

FIXME unfinished section

3.1.2 MIME

[8] [9] [10] [11] [12]

3.1.2.1 S/MIME

[30]

3.1.2.2 PGP/MIME

[29]

3.1.3 DNS

[4]

3.1.3.1 DNSSEC

[19]

3.1.3.2 Sender Policy Framework

[36] [16]

3.1.3.3 Sender ID

[35]

3.1.4 Transport Protocols

3.1.4.1 IPv4

[27] [24] [32] [25] [22] [20] [21] [26, p. 3]

3.1.4.2 IPv6

[2]

3.1.4.3 TCP

3.1.5 Remote MDA protocols

3.1.5.1 POP3

[23]

3.1.5.2 IMAPv4

[1]

Chapter 4

Analysis of current situation

FIXME waiting for this text to apear

4.1 Current state of common Technology

4.1.1 Mailrouting

4.1.1.1 SMTP

4.1.1.2 LMTP

4.1.1.3 IMAP

4.1.1.4 POP3

4.1.1.5 MS-OXMAPIHTTP

4.2 Current state of available Technology

4.3 Missing Gap

4.4 Skeleton of Mails and mail transfer

Chapter 5

Designing an approach

5.1 Defining system boundaries

5.1.1 Thread model

As an adverser we assume the following attributes:

- Available founding is huge.
- can monitor all network traffic.
- Can have own mixer infrastructure.
- Is able to read, write or modify network data freely at any point of the net.

His intensions are:

- Discover message flows
- Discover message contents
- Identify users of the system

5.1.2 User model

The assumed user of the system is:

- Does care about privacy.
- Has no special computer knowhow.

- Has the ability to install a program or plugin.
- Has no cryptographic knowhow.
- Is using a device with enough calculation power to solve cryptographic tasks.

His intensions are:

- Send personal or confidential Information securely to another user

His expectations are:

- System should be easy to configure and maintain (in an ideal world: Zero touch).
- System should be fast.
- System should be reliable.
- System should work on any client he is using.
- System should not be a legal problem to him or any of his peers.

5.1.3 Mail server admin model

The assumed mail server admin of the system is:

- Does care about privacy.
- Has cosiderable computer knowhow.
- Has the ability to install a program or plugin.
- Has possibly no cryptographic knowhow.
- Does know his own mail infrastructure
- Is using a device with enough calculation power to solve cryptographic tasks.

His intensions are:

- Support his users in sending personal or confidential information securely to another user

His expectations are:

- System should be easy to configure and maintain (in an ideal world: Zero touch).
- System should be fast.
- System should be reliable.
- System should work on any client he is using.
- System should not be a legal problem to him or his company.

5.2 Basic Requirements of an approach

Chapter 6

Specifying a target solution

6.1 Blocks

6.1.1 Preamble

- identity
- Request Number (RNR)
- Identity signature
- Message key
- Shuffle

6.1.2 Routing block

- Next hop address
- RNR (must match preamble)
- Validity time
- delivery expiry
- Message checksum (optional)
- Padding (optional)

6.1.3 Address request block

6.2 Messages

6.2.1 Basecom

- Request Server keys
- Request identity (identity and key tuple)
- Request size quota (size in bytes and end time)
- Request Puzzle spec
- Request capabilities

Chapter 7

Verification of solution

7.1 User acceptance of the target system

From a perspective of a user Collected requirements to a mail system:

Requirement	cliteria	Weight
The System should transport mails fast under normal conditions	Mails should travel with at least 1MB/min	5
The System should transport mails reliable	Mails should always arrive or their status should be retrievable	9
The System should offer anonymity against spying from third parties	Neither original sender nor final destination or any part of the message content should be determinable by any part of the system except for the original sender and the final recipient.	9
The system must be easy to handle		8
The system must be easy to install	Installation should be almost a "single-click"-Thing. Details should be copied or accessed from the existing configurations.	5
continued on next page		

continued from previous page		
Requirement	cliteria	Weight

Table 7.1: User acceptance requirements

7.2 Admin acceptance of the target system

Collected requirements to a mail system from an admin perspective:

Requirement	Criteria	weight
The System should transport mails fast under normal conditions	Mails should travel with at least 10MB/min	5
The System should transport mails reliable	Mails should always arrive or their status should be retrievable	9

7.3 Possible attacks to the system

7.3.1 Generic DoS attacks

7.3.1.1 Overloading single nodes

7.3.2 Attacks on the users anonymity

7.3.3 Reputaional attacks

7.3.3.1 Misuse for sending spam

Glossary

IMAPv4 FIXME

MTA A Mail Transfer Agent. This transfer agent rroutes mails between other components. Typically an MTA receives mails from an MRA and forwardes them to a MDA or MSA. The main task of a MTA is to provide reliable queues and solid track of all mails as long as they are not forwarded to another MTA or local storage.

MTS A Mail Transfer Service. This is a set of agents which provide the functionallity tor send and receive Messages and forward them to a local or remote store.

MDA FIXME

MRA A Mail receiving Agent. This agent receives mails from a agent. Depending on the used protocol two subtypes of MRAs are available.

> ***Client MRA*** A client MRA picks up mails in the server mail storage from a remote MDA. Client MRAs usually connect thru a standard protocol which was designed for client access. Examples for such protocols are POP3 or IMAPv4
>
> ***Server MRA*** FIXME

MSA A Mail Sending Agent. This agent sends mails to a Server MRA.

MSS A Mail Storage Service. This is a set of agents providing a reliable store for local mail accounts. It also provides Interfacing which enables clients to access the users mail.

MUA A Mail User Agent. This user agent reads mails from a local storage and allows a user to read existing mails, create and modify mails.

POP3 FIXME

Privacy From the Oxford English Dictionary: “

1. The state or condition of beeing withdrawn from the society of others, or from the public intrest; seclusion. The state or condition of beeing alone,

undisturbed, or free from public attention, as a matter of choice or right; freedom from interference or intrusion.

2. Private or retired place; private apartments; places of retreat.

3. Absence or avoidance of publicity or display; a condition approaching to secrecy or concealment. Keeping of a secret.

4. A private matter, a secret; private or personal matters or relations; The private parts.

5. Intimacy, confidential relations.

6. The state of being privy to some act.

"[7, FIXME]
In this work privacy is related to definition two. Mails should be able to be handled as a virtual private place where no one knows who is talking to whom and about what or how frequent (except for directly involved people).

Bibliography

[1] M. Crispin. *RFC3501 INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. IETF, 2003. URL: `http://tools.ietf.org/pdf/rfc3501.pdf` (cit. on p. 16).

[2] S. Deering and R. Hinden. *RFC2460 Internet Protocol, Version 6 (IPv6) Specification*. IETF, 1983. URL: `http://tools.ietf.org/pdf/rfc2460.pdf` (cit. on p. 16).

[3] s. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka. *RFC2311 S/MIME Version 2 Message Specification*. IETF, 1998. URL: `http://tools.ietf.org/pdf/rfc2311.pdf` (cit. on p. 14).

[4] D. Eastlake, E. Brunner-Williams, and B. Manning. *BCP42 Domain Name System (DNS) IANA Considerations*. IETF, 2000. URL: `http://tools.ietf.org/pdf/rfc2929.pdf` (cit. on p. 15).

[5] Temporary Committee on the ECHELON Interception System. *REPORT on the existence of a global system for the interception of private and commercial communications (ECHELON interception system)*. 2001. URL: `http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//NONSGML+REPORT+A5-2001-0264+0+DOC+PDF+V0//EN&language=EN` (cit. on p. 1).

[6] M. Elkins. *RFC2015 MIME Security with Pretty Good Privacy (PGP)*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2015.pdf` (cit. on p. 14).

[7] FIXME. *Oxford English Dictionary* (cit. on p. 28).

[8] N. Freed and N. Borenstein. *RFC2045 Multipurpose Internet Mail Extensions; (MIME) Part One: Format of Internet Message Bodies*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2045.pdf` (cit. on p. 15).

[9] N. Freed and N. Borenstein. *RFC2046 Multipurpose Internet Mail Extensions; (MIME) Part Two: Media Types*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2046.pdf` (cit. on p. 15).

[10] N. Freed and N. Borenstein. *RFC2047 Multipurpose Internet Mail Extensions; (MIME) Part Three: Message Header Extensions for Non-ASCII Text*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2046.pdf` (cit. on p. 15).

[11] N. Freed, J. Klensin, and J. Postel. *RFC2048 Multipurpose Internet Mail Extensions; (MIME) Part Four: Registration Procedures*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2048.pdf` (cit. on p. 15).

[12] N. Freed, J. Klensin, and J. Postel. *RFC2049 Multipurpose Internet Mail Extensions; (MIME) Part Five: Conformance Criteria and Examples*. IETF, 1996. URL: `http://tools.ietf.org/pdf/rfc2049.pdf` (cit. on p. 15).

[13] J. Galvin, S. Murphy, S. Crocker, and N. Freed. *RFC1847 Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted*. IETF, 1995. URL: `http://tools.ietf.org/pdf/rfc1847.pdf` (cit. on p. 14).

[14] R. Gellens and J. Klensin. *STD72 Message Submission for Mail*. IETF, 2011. URL: `http://tools.ietf.org/pdf/rfc6409.pdf` (cit. on p. 14).

[15] P. Hoffman. *RFC3207 SMTP Service Extension for Secure SMTP over Transport Layer Security*. IETF, 2002. URL: `http://tools.ietf.org/pdf/rfc3207.pdf` (cit. on p. 14).

[16] S. Kitterman. *RFC6652 Sender Policy Framework (SPF) Authentication Failure Reporting Using the Abuse Reporting Format*. IETF, 2012. URL: `http://tools.ietf.org/pdf/rfc6652.pdf` (cit. on p. 16).

[17] J. Klensin. *RFC5321 Simple Mail Transfer Protocol*. IETF, 2008. URL: `http://tools.ietf.org/pdf/rfc5321.pdf` (cit. on pp. 1, 5, 13).

[18] J. Klensin, N. Freed, and K. Moore. *RFC1870 SMTP Service Extension for Message Size Declaration*. IETF, 1995. URL: `http://tools.ietf.org/pdf/rfc1870.pdf` (cit. on p. 14).

[19] B. Laurie, G. Sisson, R. Arends, and D. Blacka. *RFC5155 DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*. IETF, 2008. URL: `http://tools.ietf.org/pdf/rfc5155.pdf` (cit. on p. 15).

[20] J. Mogul. *RFC922 BROADCASTING INTERNET DATAGRAMS IN THE PRESENCE OF SUBNETS*. IETF, 1984. URL: `http://tools.ietf.org/pdf/rfc922.pdf` (cit. on p. 16).

[21] J. Mogul and J. Postel. *RFC950 Internet Standard Subnetting Procedure*. IETF, 1985. URL: `http://tools.ietf.org/pdf/rfc950.pdf` (cit. on p. 16).

[22] Jeffrey Mogul. *RFC919 BROADCASTING INTERNET DATAGRAMS*. IETF, 1984. URL: `http://tools.ietf.org/pdf/rfc919.pdf` (cit. on p. 16).

[23] J. Myers and M. Rose. *RFC1939 Post Office Protocol - Version 3*. IETF, 1996. URL: http://tools.ietf.org/pdf/rfc1939.pdf (cit. on p. 16).

[24] J. Postel. *RFC791 INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION*. IETF, 1981. URL: http://tools.ietf.org/pdf/rfc791.pdf (cit. on p. 16).

[25] J. Postel. *RFC792 INTERNET CONTROL MESSAGE PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION*. IETF, 1981. URL: http://tools.ietf.org/pdf/rfc792.pdf (cit. on p. 16).

[26] J. Postel. *RFC793 TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION*. IETF, 1981. URL: http://tools.ietf.org/pdf/rfc793.pdf (cit. on p. 16).

[27] Jon Postel. *RFC760 DOD STANDARD INTERNET PROTOCOL*. IETF, 1980. URL: http://tools.ietf.org/pdf/rfc760.pdf (cit. on p. 16).

[28] Jonathan B. Postel. *RFC821 Simple Mail Transfer Protocol*. IETF, 1982. URL: http://tools.ietf.org/pdf/rfc821.pdf (cit. on p. 2).

[29] B. Ramsdell. *RFC2440 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*. IETF, 2004. URL: http://tools.ietf.org/pdf/rfc2440.pdf (cit. on p. 15).

[30] B. Ramsdell. *RFC3851 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*. IETF, 2004. URL: http://tools.ietf.org/pdf/rfc3851.pdf (cit. on p. 15).

[31] *RFC1087 Ethics and the Internet*. IETF, 1989. URL: http://tools.ietf.org/pdf/rfc1087.pdf (cit. on p. 11).

[32] T. Socolofsky and C. Kale. *RFC1180 A TCP/IP Tutorial*. IETF, 1991. URL: http://tools.ietf.org/pdf/rfc1180.pdf (cit. on p. 16).

[33] Wikipedia. *Anonymous remailer — Wikipedia, The Free Encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Anonymous_remailer&oldid=584455506 (cit. on p. 10).

[34] Wikipedia. *Edward Snowden — Wikipedia, The Free Encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Edward_Snowden&oldid=586147644 (cit. on p. 1).

[35] N. Williams. *RFC4401 Sender ID: Authenticating E-Mail*. IETF, 2006. URL: http://tools.ietf.org/pdf/rfc4401.pdf (cit. on p. 16).

[36] N. Williams. *RFC4408 Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*. IETF, 2006. URL: http://tools.ietf.org/pdf/rfc4408.pdf (cit. on p. 16).

Index