Protocol

This chapter describes the remail protocol internals.

Mail clients and mail servers

In order to send mail between two nodes, there ideally should be some kind of client/server setup in which the exchange occurs.

The mail client is responsible for obtaining mail from the mailbox (see next chapter) while the mail server is responsible for receiving incoming mail and storing it in the mailbox so that it may be fetched by an external client.

When transmitting mail, the mail client is to complete a proof of work computation which must have acceptable results in order for transmission to complete.

Mailboxes and the Blobchain

A mailbox can be seen as a directory containing a number of content addressed files, each of which represent an individual unit of mail. In other words, the filename of each mail file is the SHA256 hash of its content which is to contain a timestamp at the very beginning, followed by two newlines and the mail body.

A mail unit is to be treated as an immutable envelope containing metadata followed by the mail contents. To reduce filesystem pressure, each mail unit lives within a directory named after the first two bytes of the hash (xx/xxxxx...).

To prevent tampering of the mailbox, each mail unit contains references to the next and previous mail unit in the form of a mailbox name, delimited by a ':' and followed by the mail unit's content addressed hash:

All mail units are to follow the given format:

<ASCII TIMESTAMP>
<MAILBOX>:<PREVIOUS HASH IN BINARY>
<MAILBOX>:<NEXT HASH IN BINARY>
\n\n
<CONTENT>

Attempting to alter the contents of any mail unit may result in data loss and cause affected units to be rejected by the mail-server and/or client.

Sender-side Proof-of-Work

In order for a sender to transmit mail to a recipient, it must complete a proof of work algorithm at the recipient servers configured difficulty. Once the blob has been constructed on the sender-side, the sender is to monotonically sweep the nonce field, re-hashing the blob per every change. Once the blob hashes to a value with enough leading zeros that matches the servers configured difficulty, the blob will be accepted by the server. It is mandatory for the server to also verify nonce in order to reject mail with potential deficiencies.