The Cipher Mail Transport Protocol Cryptography

Jonathan Moroney

March 26, 2016

1 Overview

The Cipher Mail Transport Protocol (CMTP) has been designed to easily adapt to new crypto-systems as the need arises. Each communication from a server has a version field which denotes the crypto-system in use. There are two currently defined crypto-systems:

- 0 - Plaintext. That is, no encryption.
- 1 - X25519 / XSalsa20 with 256 bit keys. Further details below.

As the state of cryptography evolves additional crypto systems can be added.

2 Crypto System 0

As the name implies this is the 'no encryption' encryption system. There's not much to say about this system other than it should be avoided where possible. It has been included for a few reasons.

1. Someone somewhere is eventually going to want a plaintext pathway, so defining it at the protocols birth may help later on.
2. Backward compatibility with SMTP based system necessitates going plain and so having this system will allow for that compatibility to be added.
3. Having a crypto system with no keys may allow for some useful overloading with CMTP commands.

3 Crypto System 1

The choice for crypto-system 1 to use X25519 / XSalsa20 and the signing and securing algorithms was a fundamentally pragmatic one. In particular messages are enciphered with the LibSodium crypto_box_seal() [2] function. This function does the job of encrypting and signing the message as

well as throwing errors should something not match up. The signing function is the crypto_sign_detached() function [1]. Many large players on the internet are using this library and these algorithms which helped make this decision easier to make. However, this does end up meaning that the "correct" implementation of encryption is whatever LibSodium does and I don't feel good in punting like that. That said, outsourcing to LibSodium allows for this project to focus on passing messages rather than being stuck in the woods developing crypto system 1.

References

[1] TEAM, L. Public signatures. `https://download.libsodium.org/doc/public-key_cryptography/public-key_signatures.html`. Accessed: 2016-3-26.

[2] TEAM, L. Sealed boxes. `https://download.libsodium.org/doc/public-key_cryptography/sealed_boxes.html`. Accessed: 2016-3-26.