Mathematical Mesh 3.0 Part VI: Mesh Presentation Layer Security

Mesh Discovery Service

<series>draft-hallambaker-mesh-presentation

<status>informational

<stream>independent

<ipr>trust200902

<author>Phillip Hallam-Baker

<surname>Hallam-Baker

<initials>P. M.

<firstname>Phillip

<email>phill@hallambaker.com

<organization>ThresholdSecrets.com

<keyword>Threshold Cryptography

<keyword>Elliptic Curve

<keyword>Threshold Encryption

<keyword>Threshold Key Generation

<keyword>Ceremony

A presentation layer suitable for use in conjunction with HTTP and UDP transports is described.

Discussion of this draft should take place on the MathMesh mailing list (mathmesh@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/mathmesh/>.

# Introduction

## Requirements

Confidentiality

Integrity

Data reassembly

## Packet Format

Communications

## Connection and Stream Identifiers

One key exchange results in creation of a connection. This is bound to a unique set of device credentials end to end

A given UDP port receives input on multiple streams from multiple sources

Stream identifiers are unique across all resources.

A stream only contains messages for a single service (at present?)

A connection typically map to multiple streams

Creating new steams within a connection is easy.

## Stream characteristics

Multiple stream characters are supported.

Transactional - request/response. Glue multiple request/response packets together. Only need strong integrity check on final packet (AEAD tag).

Framed - similar to transactional but no requirement for a response.

Bulk - stream same as traditional TCP but with periodic strong integrity checks.

## Connection characteristics

Can define connections with traffic analysis resistance, stream Id is masked using encryption at cost of 8 bytes per packet for the stream ID plus n\*16 bytes for stream tag identifiers.

# Definitions

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in <norm="RFC2119"/>.

## Defined Terms

## Related Specifications

## Implementation Status

The implementation status of the reference code base is described in the companion document <info="draft-hallambaker-mesh-developer"/>.

# Architecture

## Credentials

## Packet Format

### Data

### Connection

# Connection Establishment

* Host credential MAY be passed plaintext
* Client credential MUST be passed encrypted.
* Mezzanine requires [Host Credential, Host Key ID, Client Ephemeral]
* Mutual requires Mezzanine + [Client Credential, Client Key ID, Host Ephemeral]
* Client Key ID before Host Key ID before ephemerals before credentials before challenge / response
* Payload always on the innermost packet
* Client ephemeral and Host ID are only sent by client
* Host ephemeral and Client ID are only sent by host
* Only require the host to maintain state after the challenge/response has been completed.

## Initial Immediate

### Client Initial

Plaintext

Extensions: { Client Ephemeral }\*

Payload

### Host Exchange

Plaintext

Host Key ID

Extensions: { Host Credential}

Mezzanine [Client Ephemeral/Host Key ID]

Extensions: { Host Ephemeral }\*

Payload

### Client Complete

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Extensions: { Client Credential}

Encrypted

Payload

## Initial Deferred

### Client Initial

**Plaintext**

Extensions: { Client Ephemeral }\*

Payload

### Host Challenge

Plaintext

Extensions: { Host Ephemeral }\*

Extensions: { Host Credential }

Extensions: { Challenge }\*

Payload

### Client Complete Deferred

**Plaintext**

Host Key ID

Client Ephemeral

Extensions: {Response}\*

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Extensions: { Client Credential }

Encrypted

Payload

## Client Immediate

### Client Exchange

**Plaintext**

Host Key ID

Client Ephemeral

Mezzanine [Client Ephemeral/Host Key ID]

Extensions: { Client Credential }

Payload

### Host Complete

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Host Ephemeral

Encrypted

Payload

## Client Deferred

Client begins with Host Credential

### Client Exchange

**Plaintext**

Host Key ID

Client Ephemeral

Mezzanine [Client Ephemeral/Host Key ID]

Extensions: { Client Credential }

Payload

### Host Challenge

Plaintext

Extensions: { Host Ephemeral }\*

Extensions: { Challenge}\*

Payload

### Client Complete Deferred

**Plaintext**

Host Key ID

Client Ephemeral

Extensions: {Response\*}

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Extensions: { Client Credential}

Encrypted

Payload

# Data Encoding

Pad all blocks to MTA

No nested length specifiers

## Packet Identifier

## String

## Binary data

## Extension set

## Outer Encrypted

## Inner Encrypted

# Packet Descriptions

## Client Initial

**Plaintext**

Extensions: { Client Ephemeral }\*

Payload

## Client Complete

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Extensions: { Client Credential}

Encrypted

Payload

## Client Complete Deferred

**Plaintext**

Host Key ID

Client Ephemeral

Extensions: {Response}\*

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Extensions: { Client Credential }

Encrypted

Payload

## Client Exchange

**Plaintext**

Host Key ID

Client Ephemeral

Mezzanine [Client Ephemeral/Host Key ID]

Extensions: { Client Credential }

Payload

## Host Exchange

Plaintext

Host Key ID

Extensions: { Host Credential}

Mezzanine [Client Ephemeral/Host Key ID]

Extensions: { Host Ephemeral }\*

Payload

## Host Challenge

Plaintext

Extensions: { Host Ephemeral }\*

Extensions: { Host Credential } [If required]

Extensions: { Challenge}\*

Payload

## Host Complete

Mezzanine [Client Ephemeral/Host Key ID]

Client Key ID

Host Ephemeral

Encrypted

Payload

# Security Considerations

# IANA Considerations

This document requires no IANA actions.

# Acknowledgements