

Workgroup: Network Working Group

Internet-Draft:

draft-hallambaker-mesh-security

Published: 2 November 2020

Intended Status: Informational

Expires: 6 May 2021

Authors: P. M. Hallam-Baker

ThresholdSecrets.com

Mathematical Mesh 3.0 Part VII: Security Considerations

Abstract

The Mathematical Mesh 'The Mesh' is an end-to-end secure infrastructure that facilitates the exchange of configuration and credential data between multiple user devices. The core protocols of the Mesh are described with examples of common use cases and reference data.

[Note to Readers]

Discussion of this draft takes place on the MATHMESH mailing list (mathmesh@ietf.org), which is archived at https://mailarchive.ietf.org/arch/search/?email_list=mathmesh.

This document is also available online at <http://mathmesh.com/Documents/draft-hallambaker-mesh-security.html>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

- [1. Introduction](#)
- [2. Definitions](#)
 - [2.1. Requirements Language](#)
 - [2.2. Defined Terms](#)
 - [2.3. Related Specifications](#)
 - [2.4. Implementation Status](#)
 - [2.5. Shared Classes](#)
 - [2.5.1. Classes describing keys](#)
 - [2.5.2. Structure: KeyData](#)
 - [2.5.3. Structure: CompositePrivate](#)
 - [2.6. Assertion classes](#)
 - [2.6.1. Structure: Assertion](#)
 - [2.6.2. Structure: Condition](#)
 - [2.6.3. Base Classes](#)
 - [2.6.4. Structure: Connection](#)
 - [2.6.5. Structure: Activation](#)
 - [2.6.6. Structure: ActivationEntry](#)
 - [2.6.7. Mesh Profile Classes](#)
 - [2.6.8. Structure: Profile](#)
 - [2.6.9. Structure: ProfileDevice](#)
 - [2.6.10. Structure: ProfileAccount](#)
 - [2.6.11. Structure: ProfileUser](#)
 - [2.6.12. Structure: ProfileGroup](#)
 - [2.6.13. Structure: ProfileService](#)
 - [2.6.14. Structure: ProfileHost](#)
 - [2.6.15. Connection Assertions](#)
 - [2.6.16. Structure: ConnectionDevice](#)
 - [2.6.17. Structure: ConnectionApplication](#)
 - [2.6.18. Structure: ConnectionGroup](#)
 - [2.6.19. Structure: ConnectionService](#)
 - [2.6.20. Structure: ConnectionHost](#)
 - [2.6.21. Activation Assertions](#)
 - [2.6.22. Structure: ActivationDevice](#)
 - [2.6.23. Structure: ActivationAccount](#)
 - [2.6.24. Structure: ActivationApplication](#)
 - [2.7. Data Structures](#)
 - [2.7.1. Structure: Contact](#)
 - [2.7.2. Structure: Anchor](#)
 - [2.7.3. Structure: TaggedSource](#)
 - [2.7.4. Structure: ContactGroup](#)

- [2.7.5. Structure: ContactPerson](#)
- [2.7.6. Structure: ContactOrganization](#)
- [2.7.7. Structure: OrganizationName](#)
- [2.7.8. Structure: PersonName](#)
- [2.7.9. Structure: NetworkAddress](#)
- [2.7.10. Structure: NetworkProtocol](#)
- [2.7.11. Structure: Role](#)
- [2.7.12. Structure: Location](#)
- [2.7.13. Structure: Bookmark](#)
- [2.7.14. Structure: Reference](#)
- [2.7.15. Structure: Task](#)
- [2.8. Catalog Entries](#)
 - [2.8.1. Structure: CatalogedEntry](#)
 - [2.8.2. Structure: CatalogedDevice](#)
 - [2.8.3. Structure: CatalogedPublication](#)
 - [2.8.4. Structure: CatalogedCredential](#)
 - [2.8.5. Structure: CatalogedNetwork](#)
 - [2.8.6. Structure: CatalogedContact](#)
 - [2.8.7. Structure: CatalogedAccess](#)
 - [2.8.8. Structure: CryptographicCapability](#)
 - [2.8.9. Structure: CapabilityDecrypt](#)
 - [2.8.10. Structure: CapabilityDecryptPartial](#)
 - [2.8.11. Structure: CapabilityDecryptServiced](#)
 - [2.8.12. Structure: CapabilitySign](#)
 - [2.8.13. Structure: CapabilityKeyGenerate](#)
 - [2.8.14. Structure: CapabilityFairExchange](#)
 - [2.8.15. Structure: CatalogedBookmark](#)
 - [2.8.16. Structure: CatalogedTask](#)
 - [2.8.17. Structure: CatalogedApplication](#)
 - [2.8.18. Structure: CatalogedMember](#)
 - [2.8.19. Structure: CatalogedGroup](#)
 - [2.8.20. Structure: CatalogedApplicationSSH](#)
 - [2.8.21. Structure: CatalogedApplicationMail](#)
 - [2.8.22. Structure: CatalogedApplicationNetwork](#)
- [2.9. Publications](#)
 - [2.9.1. Structure: DevicePreconfiguration](#)
- [2.10. Messages](#)
 - [2.10.1. Structure: Message](#)
 - [2.10.2. Structure: MessageError](#)
 - [2.10.3. Structure: MessageComplete](#)
 - [2.10.4. Structure: MessagePinValidated](#)
 - [2.10.5. Structure: MessagePin](#)
 - [2.10.6. Structure: RequestConnection](#)
 - [2.10.7. Structure: AcknowledgeConnection](#)
 - [2.10.8. Structure: RespondConnection](#)
 - [2.10.9. Structure: MessageContact](#)
 - [2.10.10. Structure: GroupInvitation](#)
 - [2.10.11. Structure: RequestConfirmation](#)
 - [2.10.12. Structure: ResponseConfirmation](#)

- [2.10.13. Structure: RequestTask](#)
 - [2.10.14. Structure: MessageClaim](#)
 - [2.10.15. Structure: ProcessResult](#)
- 3. [Mesh Portal Service Reference](#)
 - 3.1. [Request Messages](#)
 - 3.1.1. [Message: MeshRequest](#)
 - 3.2. [Response Messages](#)
 - 3.2.1. [Message: MeshResponse](#)
 - 3.3. [Imported Objects](#)
 - 3.4. [Common Structures](#)
 - 3.4.1. [Structure: KeyValue](#)
 - 3.4.2. [Structure: SearchConstraints](#)
 - 3.5. [Transaction: Hello](#)
 - 3.6. [Transaction: ValidateAccount](#)
 - 3.6.1. [Message: ValidateRequest](#)
 - 3.6.2. [Message: ValidateResponse](#)
 - 3.7. [Transaction: CreateAccount](#)
 - 3.7.1. [Message: CreateRequest](#)
 - 3.7.2. [Message: CreateResponse](#)
 - 3.8. [Transaction: DeleteAccount](#)
 - 3.8.1. [Message: DeleteRequest](#)
 - 3.8.2. [Message: DeleteResponse](#)
 - 3.9. [Transaction: Get](#)
 - 3.9.1. [Message: GetRequest](#)
 - 3.9.2. [Message: GetResponse](#)
 - 3.10. [Transaction: Publish](#)
 - 3.10.1. [Message: PublishRequest](#)
 - 3.10.2. [Message: PublishResponse](#)
 - 3.11. [Transaction: Status](#)
 - 3.11.1. [Message: StatusRequest](#)
 - 3.11.2. [Message: StatusResponse](#)
 - 3.12. [Transaction: ConnectStart](#)
 - 3.12.1. [Message: ConnectStartRequest](#)
 - 3.12.2. [Message: ConnectStartResponse](#)
 - 3.13. [Transaction: ConnectStatus](#)
 - 3.13.1. [Message: ConnectStatusRequest](#)
 - 3.13.2. [Message: ConnectStatusResponse](#)
 - 3.14. [Transaction: ConnectPending](#)
 - 3.14.1. [Message: ConnectPendingRequest](#)
 - 3.14.2. [Message: ConnectPendingResponse](#)
 - 3.15. [Transaction: ConnectComplete](#)
 - 3.15.1. [Message: ConnectCompleteRequest](#)
 - 3.15.2. [Message: ConnectCompleteResponse](#)
 - 3.16. [Transaction: Transfer](#)
 - 3.16.1. [Message: TransferRequest](#)
 - 3.16.2. [Message: TransferResponse](#)
- 4. [Assets](#)
 - 4.1. [Data](#)
 - 4.2. [Credentials](#)

- [4.3. Reputation](#)
 - [4.3.1. Outbound Messaging Abuse \(\)](#)
- [5. Risks](#)
 - [5.1. Confidentiality](#)
 - [5.1.1. Privacy](#)
 - [5.2. Integrity](#)
 - [5.3. Availability](#)
 - [5.3.1. Data loss](#)
 - [5.3.2. Partial data survivability](#)
 - [5.4. Inbound Messaging Abuse \(Spam\)](#)
- [6. Threats](#)
 - [6.1. End point Compromise](#)
- [7. Controls](#)
 - [7.1. Cryptographic](#)
 - [7.1.1. Triple lock](#)
 - [7.1.2. Key Protection](#)
 - [7.1.3. Key and Nonce Generation](#)
 - [7.1.4. Key Escrow and Recovery](#)
 - [7.1.5. Profile Verification](#)
 - [7.1.6. Identity Validation](#)
 - [7.1.7. Trust Broker Accountability](#)
 - [7.2. Mesh Messaging](#)
 - [7.2.1. Ingress Control](#)
 - [7.2.2. Egress Control](#)
 - [7.2.3. Security Signal](#)
 - [7.2.4. Accountability](#)
- [8. Security Considerations](#)
 - [8.1. Integrity](#)
 - [8.1.1. DNS Spoofing](#)
 - [8.1.2. TLS Downgrade](#)
 - [8.1.3. TLS Service Impersonation](#)
 - [8.1.4. Request Replay Attack](#)
 - [8.1.5. Response Replay Attack](#)
 - [8.2. Confidentiality](#)
 - [8.2.1. Side Channel Attack](#)
 - [8.2.2. Session Key Leakage](#)
- [9. IANA Considerations](#)
- [10. Acknowledgements](#)
- [11. Normative References](#)
- [12. Informative References](#)

1. Introduction

2. 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.

2.1. 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 [[RFC2119](#)].

2.2. Defined Terms

The terms of art used in this document are described in the *Mesh Architecture Guide* [[draft-hallambaker-mesh-architecture](#)].

2.3. Related Specifications

The architecture of the Mathematical Mesh is described in the *Mesh Architecture Guide* [[draft-hallambaker-mesh-architecture](#)]. The Mesh documentation set and related specifications are described in this document.

2.4. Implementation Status

The implementation status of the reference code base is described in the companion document [[draft-hallambaker-mesh-developer](#)].

2.5. Shared Classes

The following classes are used as common elements in Mesh profile specifications.

2.5.1. Classes describing keys

2.5.2. Structure: KeyData

The KeyData class is used to describe public key pairs and trust assertions associated with a public key.

Udf: String (Optional) UDF fingerprint of the public key parameters

X509Certificate: Binary (Optional) List of X.509 Certificates

X509Chain: Binary [0..Many] X.509 Certificate chain.

X509CSR: Binary (Optional) X.509 Certificate Signing Request.

NotBefore: DateTime (Optional) If present specifies a time instant that use of the private key is not valid before.

NotOnOrAfter: DateTime (Optional) If present specifies a time instant that use of the private key is not valid on or after.

2.5.3. Structure: CompositePrivate

Inherits: Key

DeviceKeyUdf: String (Optional) UDF fingerprint of the bound device key (if used).

2.6. Assertion classes

Classes that are derived from an assertion.

2.6.1. Structure: Assertion

Parent class from which all assertion classes are derived

Names: String [0..Many] Fingerprints of index terms for profile retrieval. The use of the fingerprint of the name rather than the name itself is a precaution against enumeration attacks and other forms of abuse.

Updated: DateTime (Optional) The time instant the profile was last modified.

NotaryToken: String (Optional) A Uniform Notary Token providing evidence that a signature was performed after the notary token was created.

2.6.2. Structure: Condition

Parent class from which all condition classes are derived.

[No fields]

2.6.3. Base Classes

Abstract classes from which the Profile, Activation and Connection classes are derived.

2.6.4. Structure: Connection

Inherits: Assertion UDF of the connection target.

SubjectUdf: String (Optional)

AuthorityUdf: String (Optional) UDF of the connection source.

2.6.5. Structure: Activation

Inherits: Assertion

Contains the private activation information for a Mesh application running on a specific device

ActivationKey: String (Optional) Secret seed used to derive keys that are not explicitly specified.

Entries: ActivationEntry [0..Many] Activation of named resources.

2.6.6. Structure: `ActivationEntry`

Resource: `String` (Optional) Name of the activated resource

Key: `KeyData` (Optional) The activation key or key share

2.6.7. Mesh Profile Classes

Classes describing Mesh Profiles. All Profiles are Assertions derived from `Assertion`.

2.6.8. Structure: `Profile`

Inherits: `Assertion`

Parent class from which all profile classes are derived

ProfileSignature: `KeyData` (Optional) The permanent signature key used to sign the profile itself. The UDF of the key is used as the permanent object identifier of the profile. Thus, by definition, the `KeySignature` value of a `Profile` does not change under any circumstance.

2.6.9. Structure: `ProfileDevice`

Inherits: `Profile`

Describes a mesh device.

Description: `String` (Optional) Description of the device

BaseEncryption: `KeyData` (Optional) Base key contribution for encryption keys. Also used to decrypt activation data sent to the device during connection to an account.

BaseAuthentication: `KeyData` (Optional) Base key contribution for authentication keys. Also used to authenticate the device during connection to an account.

BaseSignature: `KeyData` (Optional) Base key contribution for signature keys.

2.6.10. Structure: `ProfileAccount`

Base class for the account profiles `ProfileUser` and `ProfileGroup`. These subclasses may be merged at some future date.

Inherits: `Profile` The account address. This is either a DNS service
AccountAddress: `String` (Optional) address (e.g. `alice@example.com`) or a Mesh Name (`@alice`).

ServiceUdf: String (Optional)

The fingerprint of the service profile to which the account is currently bound.

EscrowEncryption: KeyData (Optional) Escrow key associated with the account.

AccountEncryption: KeyData (Optional) Key currently used to encrypt data under this profile

AdministratorSignature: KeyData (Optional) Key used to sign connection assertions to the account.

2.6.11. Structure: ProfileUser

Inherits: ProfileAccount

Account assertion. This is signed by the service hosting the account.

AccountAuthentication: KeyData (Optional) Key used to authenticate requests made under this user account.

AccountSignature: KeyData (Optional) Key used to sign data under the account.

2.6.12. Structure: ProfileGroup

Inherits: ProfileAccount

Describes a group. Note that while a group is created by one person who becomes its first administrator, control of the group may pass to other administrators over time.

[No fields]

2.6.13. Structure: ProfileService

Inherits: Profile

Profile of a Mesh Service

ServiceAuthentication: KeyData (Optional) Key used to authenticate service connections.

ServiceEncryption: KeyData (Optional) Key used to encrypt data under this profile

ServiceSignature: KeyData (Optional) Key used to sign data under the account.

2.6.14. Structure: ProfileHost

Inherits: Profile

KeyAuthentication: KeyData (Optional)

Key used to authenticate service connections.

KeyEncryption: KeyData (Optional) Key used to pass encrypted data to the device such as a

2.6.15. Connection Assertions

Connection assertions are used to authenticate and authorize interactions between devices and the service currently servicing the account. They SHOULD NOT be visible to external parties.

2.6.16. Structure: ConnectionDevice

Inherits: Connection

Connection assertion used to authenticate service requests made by a device.

AccountAddress: String (Optional) The account address

DeviceSignature: KeyData (Optional) The signature key for use of the device under the profile

DeviceEncryption: KeyData (Optional) The encryption key for use of the device under the profile

DeviceAuthentication: KeyData (Optional) The authentication key for use of the device under the profile

2.6.17. Structure: ConnectionApplication

Inherits: Connection

Connection assertion stating that a particular device is

[No fields]

2.6.18. Structure: ConnectionGroup

Describes the connection of a member to a group.

Inherits: Connection

[No fields]

2.6.19. Structure: ConnectionService

Inherits: Connection

[No fields]

2.6.20. Structure: ConnectionHost

Inherits: Connection

[No fields]

2.6.21. Activation Assertions

2.6.22. Structure: ActivationDevice

Contains activation data for device specific keys used in the context of a Mesh account.

Inherits: Activation The UDF of the account

AccountUdf: String (Optional)

2.6.23. Structure: ActivationAccount

Inherits: Activation Grant access to profile online signing key

ProfileSignature: KeyData (Optional) used to sign updates to the profile.

AdministratorSignature: KeyData (Optional) Grant access to Profile administration key used to make changes to administrator catalogs.

AccountEncryption: KeyData (Optional) Grant access to ProfileUser account encryption key

AccountAuthentication: KeyData (Optional) Grant access to ProfileUser account authentication key

AccountSignature: KeyData (Optional) Grant access to ProfileUser account signature key

2.6.24. Structure: ActivationApplication

Inherits: Activation

[No fields]

2.7. Data Structures

Classes describing data used in cataloged data.

2.7.1. Structure: Contact

Inherits: Assertion

Base class for contact entries.

Id: String (Optional) The globally unique contact identifier.

anchors: Anchor [0..Many] Mesh fingerprints associated with the contact.

NetworkAddresses: NetworkAddress [0..Many]

Network address entries

Locations: Location [0..Many] The physical locations the contact is associated with.

Roles: Role [0..Many] The roles of the contact

Bookmark: Bookmark [0..Many] The Web sites and other online presences of the contact

Sources: TaggedSource [0..Many] Source(s) from which this contact was constructed.

2.7.2. Structure: Anchor

Trust anchor

Udf: String (Optional) The trust anchor.

Validation: String (Optional) The means of validation.

2.7.3. Structure: TaggedSource

Source from which contact information was obtained.

LocalName: String (Optional) Short name for the contact information.

Validation: String (Optional) The means of validation.

BinarySource: Binary (Optional) The contact data in binary form.

EnvelopedSource: Enveloped (Optional) The contact data in enveloped form. If present, the BinarySource property is ignored.

2.7.4. Structure: ContactGroup

Inherits: Contact

Contact for a group, including encryption groups.

[No fields]

2.7.5. Structure: ContactPerson

Inherits: Contact List of person names in order of preference

CommonNames: PersonName [0..Many]

2.7.6. Structure:

ContactOrganization

Inherits: Contact

CommonNames: OrganizationName [0..Many]

List of person names in
order of preference

2.7.7. Structure: OrganizationName

The name of an organization

Inactive: Boolean (Optional) If true, the name is not in current use.

RegisteredName: String (Optional) The registered name.

DBA: String (Optional) Names that the organization uses including trading names and doing business as names.

2.7.8. Structure: PersonName

The name of a natural person

Inactive: Boolean (Optional) If true, the name is not in current use.

FullName: String (Optional) The preferred presentation of the full name.

Prefix: String (Optional) Honorific or title, E.g. Sir, Lord, Dr., Mr.

First: String (Optional) First name.

Middle: String [0..Many] Middle names or initials.

Last: String (Optional) Last name.

Suffix: String (Optional) Nominal suffix, e.g. Jr., III, etc.

PostNominal: String (Optional) Post nominal letters (if used).

2.7.9. Structure: NetworkAddress

Provides all means of contacting the individual according to a particular network address

Inactive: Boolean (Optional)

If true, the name is not in current use.

Address: String (Optional) The network address, e.g.
alice@example.com

NetworkCapability: String [0..Many] The capabilities bound to this address.

EnvelopedProfileAccount: Enveloped (Optional) The account profile

Protocols: NetworkProtocol [0..Many] Public keys associated with the network address

2.7.10. Structure: NetworkProtocol

Protocol: String (Optional) The IANA protocol|identifier of the network protocols by which the contact may be reached using the specified Address.

2.7.11. Structure: Role

OrganizationName: String (Optional) The organization at which the role is held

Titles: String [0..Many] The titles held with respect to that organization.

Locations: Location [0..Many] Postal or physical addresses associated with the role.

2.7.12. Structure: Location

Appartment: String (Optional)

Street: String (Optional)

District: String (Optional)

Locality: String (Optional)

County: String (Optional)

Postcode: String (Optional)

Country: String (Optional)

Uri: String (Optional)

Title: String (Optional)

Role: String [0..Many]

2.7.13. Structure: Bookmark

2.7.14. Structure: Reference

MessageId: String (Optional) The received message to which this is a response

ResponseId: String (Optional) Message that was generated in response to the original (optional).

Relationship: String (Optional) The relationship type. This can be Read, Unread, Accept, Reject.

2.7.15. Structure: Task

Key: String (Optional) Unique key.

Start: DateTime (Optional) 2.8. Catalog Entries

Finish: DateTime (Optional)

StartTravel: String (Optional) 2.8.1. Structure: CatalogedEntry

FinishTravel: String (Optional)

TimeZone: String (Optional) Base class for cataloged Mesh data.

Title: String (Optional)

Description: String (Optional)

Location: String (Optional)

Trigger: String [0..Many]

Conference: String [0..Many]

Repeat: String (Optional)

Busy: Boolean (Optional)

Labels: String [0..Many] The set of labels describing the entry

2.8.2. Structure: CatalogedDevice

Inherits: CatalogedEntry

Public device entry, indexed under the device ID Hello

Udf: String (Optional)

UDF of the signature key of the device in the Mesh

DeviceUdf: String (Optional) UDF of the offline signature key of the device

SignatureUdf: String (Optional) UDF of the account online signature key

EnvelopedProfileUser: Enveloped (Optional) The Mesh profile

EnvelopedProfileDevice: Enveloped (Optional) The device profile

EnvelopedConnectionUser: Enveloped (Optional) The public assertion demonstrating connection of the Device to the Mesh

EnvelopedActivationDevice: Enveloped (Optional) The activation of the device within the Mesh account

EnvelopedActivationAccount: Enveloped (Optional) The activation of the device within the Mesh account

EnvelopedActivationApplication: Enveloped [0..Many] Application activations granted to the device.

2.8.3. Structure: CatalogedPublication

Inherits: CatalogedEntry

A publication.

Id: String (Optional) Unique identifier code

Authenticator: String (Optional) The witness key value to use to request access to the record.

EnvelopedData: DareEnvelope (Optional) Dare Envelope containing the entry data. The data type is specified by the envelope metadata.

NotOnOrAfter: DateTime (Optional) Expiration time (inclusive)

2.8.4. Structure: CatalogedCredential

Inherits: CatalogedEntry

Protocol: String (Optional) 2.8.5. Structure: CatalogedNetwork

Service: String (Optional)

Username: String (Optional)

Password: String (Optional)

Inherits: CatalogedEntry

Protocol: String (Optional)
Service: String (Optional)
Username: String (Optional) 2.8.6. Structure: CatalogedContact
Password: String (Optional)

Inherits: CatalogedEntry Unique key.
Key: String (Optional)
Self: Boolean (Optional) If true, this catalog entry is for the user who created the catalog.

2.8.7. Structure: CatalogedAccess

Inherits: CatalogedEntry
[No fields]

2.8.8. Structure: CryptographicCapability

Id: String (Optional) The identifier of the capability. If this is a user capability, MUST match the KeyData identifier. If this is a serviced capability, MUST match the value of ServiceId on the corresponding service capability.

KeyData: KeyData (Optional) The key that enables the capability

EnvelopedKeyShares: Enveloped [0..Many] One or more enveloped key shares.

SubjectId: String (Optional) The identifier of the resource that is controlled using the key.

SubjectAddress: String (Optional) The address of the resource that is controlled using the key.

2.8.9. Structure: CapabilityDecrypt

Inherits: CryptographicCapability
The corresponding key is a decryption key
[No fields]

2.8.10. Structure: CapabilityDecryptPartial

Inherits: CapabilityDecrypt
The corresponding key is an encryption key

ServiceId: String (Optional) The identifier used to claim the capability from the service.[Only present for a partial capability.]

ServiceAddress: String (Optional)

The service account that supports a serviced capability. [Only present for a partial capability.]

2.8.11. Structure: CapabilityDecryptServiced

Inherits: CapabilityDecrypt

The corresponding key is an encryption key

AuthenticationId: String (Optional) UDF of trust root under which request to use a serviced capability must be authorized. [Only present for a serviced capability]

2.8.12. Structure: CapabilitySign

Inherits: CryptographicCapability

The corresponding key is an administration key

[No fields]

2.8.13. Structure: CapabilityKeyGenerate

Inherits: CryptographicCapability

The corresponding key is a key that may be used to generate key shares.

[No fields]

2.8.14. Structure: CapabilityFairExchange

Inherits: CryptographicCapability

The corresponding key is a decryption key to be used in accordance with the Micali Fair Electronic Exchange with Invisible Trusted Parties protocol.

[No fields]

2.8.15. Structure: CatalogedBookmark

Inherits: CatalogedEntry

Uri: String (Optional)

Title: String (Optional)

Path: String (Optional)

Inherits: CatalogedEntry Unique key.

EnvelopedTask: Enveloped (Optional)

Title: String (Optional)

Key: String (Optional)

2.8.16. Structure: CatalogedTask

2.8.17. Structure:

CatalogedApplication

Inherits: CatalogedEntry
Key: String (Optional) Enveloped keys for use with Application
EnvelopedCapabilities: DareEnvelope [0..Many]
2.8.18. Structure:

CatalogedMember

ContactAddress: String (Optional)
MemberCapabilityId: String (Optional) **2.8.19. Structure:**
ServiceCapabilityId: String (Optional) CatalogedGroup
Inherits: CatalogedEntry

Inherits: CatalogedApplication The Mesh profile
EnvelopedProfileGroup: Enveloped (Optional)
EnvelopedActivationAccount: Enveloped (Optional) The activation of
the Mesh account the device within

2.8.20. Structure: CatalogedApplicationSSH

Inherits: CatalogedApplication
[No fields]

2.8.21. Structure: CatalogedApplicationMail

Inherits: CatalogedApplication
[No fields]

2.8.22. Structure: CatalogedApplicationNetwork

Inherits: CatalogedApplication
[No fields]

2.9. Publications

2.9.1. Structure: DevicePreconfiguration

A data structure that is passed

EnvelopedProfileDevice: Enveloped (Optional) The device profile

EnvelopedConnectionDevice: Enveloped (Optional) The device
connection

ConnectUri: String (Optional) The connection URI. This would
normally be printed on the device as a QR code.

2.10. Messages

2.10.1. Structure: Message

MessageId: String (Optional)

Unique per-message ID. When encapsulating a Mesh Message in a DARE envelope, the envelope EnvelopeID field MUST be a UDF fingerprint of the MessageId value.

Sender: String (Optional) 2.10.2. **Structure: MessageError**
Recipient: String (Optional)

Inherits: Message

ErrorCode: String (Optional) 2.10.3. **Structure: MessageComplete**

Inherits: Message

References: Reference [0..Many] 2.10.4. **Structure:**
MessagePinValidated

Inherits: Message Enveloped data that is authenticated by means of
AuthenticatedData: DareEnvelope (Optional) the PIN

ClientNonce: Binary (Optional) Nonce provided by the client to
validate the PIN

PinId: String (Optional) Pin identifier value calculated from the
PIN code, action and account address.

PinWitness: Binary (Optional) Witness value calculated as KDF
(Device.Udf + AccountAddress, ClientNonce)

2.10.5. **Structure: MessagePin**

Account: String (Optional) If true, authentication against the PIN

Inherits: Message code is sufficient to complete the associated

Expires: DateTime (Optional) action without further authorization.

Automatic: Boolean (Optional)

SaltedPin: String (Optional) PIN code bound to the specified
action.

Action: String (Optional) The action to which this PIN code is
bound.

2.10.6. **Structure: RequestConnection**

Connection request message. This message contains the information

Inherits: MessagePinValidated

AccountAddress: String (Optional) 2.10.7. **Structure:**
AcknowledgeConnection

Connection request message generated by a service on receipt of a
valid MessageConnectionRequestClient

Inherits: Message

EnvelopedRequestConnection: Enveloped (Optional)

The client

connection request.

ServerNonce: Binary (Optional) 2.10.8. Structure:

Witness: String (Optional) RespondConnection

Respond to RequestConnection message to grant or refuse the connection request.

Inherits: Message The response to the request. One of "Accept",

Result: String (Optional) "Reject" or "Pending".

CatalogedDevice: CatalogedDevice (Optional) The device information.

MUST be present if the value of Result is "Accept". MUST be absent or null otherwise.

2.10.9. Structure: MessageContact

Inherits: MessagePinValidated If true, requests that the recipient

Reply: Boolean (Optional) return their own contact information in reply.

Subject: String (Optional) Optional explanation of the reason for the request.

PIN: String (Optional) One time authentication code supplied to a recipient to allow authentication of the response.

2.10.10. Structure: GroupInvitation

Inherits: Message

Text: String (Optional) 2.10.11. Structure: RequestConfirmation

Inherits: Message

Text: String (Optional) 2.10.12. Structure: ResponseConfirmation

Inherits: Message

Request: Enveloped (Optional) 2.10.13. Structure: RequestTask

Accept: Boolean (Optional)

Inherits: Message

[No fields]

2.10.14. Structure: MessageClaim

Inherits: Message

PublicationId: String (Optional)

ServiceAuthenticate: String (Optional)

DeviceAuthenticate: String (Optional)

Expires: DateTime (Optional)

2.10.15. Structure: ProcessResult

For future use, allows logging of operations and results

Inherits: Message The error report code.

Success: Boolean (Optional)

ErrorReport: String (Optional)

3. Mesh Portal Service Reference

HTTP Well Known Service Prefix: /.well-known/mmm

Every Mesh Portal

Service transaction consists of exactly one request followed by exactly one response. Mesh Service transactions MAY cause modification of the data stored in the Mesh Portal or the Mesh itself but do not cause changes to the connection state. The protocol itself is thus idempotent. There is no set sequence in which operations are required to be performed. It is not necessary to perform a Hello transaction prior to a ValidateAccount, Publish or any other transaction.

3.1. Request Messages

A Mesh Portal Service request consists of a payload object that inherits from the MeshRequest class. When using the HTTP binding, the request MUST specify the portal DNS address in the HTTP Host field.

3.1.1. Message: MeshRequest

Base class for all request messages.

Portal: String (Optional) Name of the Mesh Portal Service to which the request is directed.

3.2. Response Messages

A Mesh Portal Service response consists of a payload object that inherits from the MeshResponse class. When using the HTTP binding, the response SHOULD report the Status response code in the HTTP response message. However the response code returned in the payload object MUST always be considered authoritative.

3.2.1. Message: MeshResponse

Base class for all response messages. Contains only the status code and status description fields.

[No fields]

3.3. Imported Objects

The Mesh Service protocol makes use of JSON objects defined in the JOSE Signature and Encryption specifications.

3.4. Common Structures

The following common structures are used in the protocol messages:

3.4.1. Structure: KeyValue

Describes a Key/Value structure used to make queries for records matching one or more selection criteria.

Key: String (Optional) The data retrieval key.

Value: String (Optional) The data value to match.

3.4.2. Structure: SearchConstraints

Specifies constraints to be applied to a search result. These allow a client to limit the number of records returned, the quantity of data returned, the earliest and latest data returned, etc.

NotBefore: DateTime (Optional) Only data published on or after the specified time instant is requested.

Before: DateTime (Optional) Only data published before the specified time instant is requested. This excludes data published at the specified time instant.

MaxEntries: Integer (Optional) Maximum number of data entries to return.

MaxBytes: Integer (Optional) Maximum number of data bytes to return.

PageKey: String (Optional) Specifies a page key returned in a previous search operation in which the number of responses exceeded the specified bounds.

When a page key is specified, all the other search parameters except for MaxEntries and MaxBytes are ignored and the service returns the next set of data responding to the earlier query.

3.5. Transaction: Hello

Request: HelloRequest

Response: HelloResponse Report service and version information.

The Hello transaction provides a means of determining which protocol versions, message encodings and transport protocols are supported by the service.

3.6. Transaction: ValidateAccount

Request: `ValidateRequest`

Response: `ValidateResponse` Request validation of a proposed name for a new account.

For validation of a user's account name during profile creation.

3.6.1. Message: ValidateRequest

Inherits: `MeshRequest`

Describes the proposed account properties. Currently, these are limited to the account name but could be extended in future versions of the protocol.

Account: `String (Optional)` Account name requested

Reserve: `Boolean (Optional)` If true, request a reservation for the specified account name. Note that the service is not obliged to honor reservation requests.

Language: `String [0..Many]` List of ISO language codes in order of preference. For creating explanatory text.

3.6.2. Message: ValidateResponse

Inherits: `MeshResponse`

States whether the proposed account properties are acceptable and (optional) returns an indication of what properties are valid.

Note that receiving a 'Valid' response to a Validate Request does not guarantee creation of the account. In addition to the possibility that the account name could be requested by another user between the Validate and Create transactions, a portal service MAY perform more stringent validation criteria when an account is actually being created. For example, checking with the authoritative list of current accounts rather than a cached copy.

Valid: Boolean (Optional)

If true, the specified account identifier is acceptable. If false, the account identifier is rejected.

Minimum: Integer (Optional) Specifies the minimum length of an account name.

Maximum: Integer (Optional) Specifies the maximum length of an account name.

InvalidCharacters: String (Optional) A list of characters that the service does not accept in account names. The list of characters MAY not be exhaustive but SHOULD include any illegal characters in the proposed account name.

Reason: String (Optional) Text explaining the reason an account name was rejected.

3.7. Transaction: CreateAccount

Request: CreateRequest

Response: CreateResponse Request creation of a new portal account.

Unlike a profile, a mesh account is specific to a particular Mesh portal. A mesh account must be created and accepted before a profile can be published.

3.7.1. Message: CreateRequest

Request creation of a new portal account. The request specifies the requested account identifier and the Mesh profile to be associated with the account.

Inherits: MeshRequest Account identifier requested.

Account: String (Optional)

3.7.2. Message: CreateResponse

Inherits: MeshResponse

Reports the success or failure of a Create transaction.

[No fields]

3.8. Transaction: DeleteAccount

Request: DeleteRequest

Response: DeleteResponse Request deletion of a portal account.

Deletes a portal account but not the underlying profile. Once registered, profiles are permanent.

3.8.1. Message: DeleteRequest

Request deletion of a new portal account. The request specifies the requested account identifier.

Inherits: MeshRequest Account identifier to be deleted.

Account: String (Optional)

3.8.2. Message: DeleteResponse

Inherits: MeshResponse

Reports the success or failure of a Delete transaction.

[No fields]

3.9. Transaction: Get

Request: GetRequest

Response: GetResponse Search for data in the mesh that matches a set of properties described by a sequence of key/value pairs.

3.9.1. Message: GetRequest

Describes the Portal or Mesh data to be retrieved.

Inherits: MeshRequest Lookup by profile ID

Identifier: String (Optional)

Account: String (Optional) Lookup by Account ID

KeyValues: KeyValue [0..Many] List of KeyValue pairs specifying the conditions to be met

SearchConstraints: SearchConstraints (Optional) Constrain the search to a specific time interval and/or limit the number and/or total size of data records returned.

Multiple: Boolean (Optional) If true return multiple responses if available

Full: Boolean (Optional) If true, the client requests that the full Mesh data record be returned containing both the Mesh entry itself and the Mesh metadata that allows the date and time of the publication of the Mesh entry to be verified.

3.9.2. Message: GetResponse

Reports the success or failure of a Get transaction. If a Mesh entry matching the specified profile is found, contains the list of entries matching the request.

Inherits: MeshResponse

DataItems: DataItem [0..Many] List of mesh data records matching the request.

PageKey: String (Optional) If non-null, indicates that the number and/or size of the data records returned exceeds either the SearchConstraints specified in the request or internal server limits.

3.10. Transaction: Publish

Request: PublishRequest

Response: PublishResponse Publish a profile or key escrow entry to the mesh.

3.10.1. Message: PublishRequest

Requests publication of the specified Mesh entry.

Inherits: MeshRequest

[No fields]

3.10.2. Message: PublishResponse

Reports the success or failure of a Publish transaction.

Inherits: MeshResponse

[No fields]

3.11. Transaction: Status

Request: StatusRequest

Response: StatusResponse Request the current status of the mesh as seen by the portal to which it is directed.

The response to the status request contains the last signed checkpoint and proof chains for each of the peer portals that have been checkpointed.

[Not currently implemented]

3.11.1. Message: StatusRequest

Inherits: MeshRequest

Initiates a status transaction.

[No fields]

3.11.2. Message: StatusResponse

Reports the success or failure of a Status transaction.

Inherits: MeshResponse Time that the last write update was made to
LastWriteTime: DateTime (Optional) the Mesh

LastCheckpointTime: DateTime (Optional) Time that the last Mesh
checkpoint was calculated.

NextCheckpointTime: DateTime (Optional) Time at which the next Mesh
checkpoint should be calculated.

CheckpointValue: String (Optional) Last checkpoint value.

3.12. Transaction: ConnectStart

Request: ConnectStartRequest

Response: ConnectStartResponse Request connection of a new device
to a mesh profile

3.12.1. Message: ConnectStartRequest

Inherits: MeshRequest
Initial device connection request.

SignedRequest: SignedConnectionRequest (Optional) Device connection
request signed by the signature key of the device requesting
connection.

AccountID: String (Optional) Account identifier of account to which
the device is requesting connection.

3.12.2. Message: ConnectStartResponse

Reports the success or failure of a ConnectStart transaction.

Inherits: MeshRequest
[No fields]

3.13. Transaction: ConnectStatus

Request: ConnectStatusRequest

Response: ConnectStatusResponse Request status of pending
connection request of a new device
to a mesh profile

3.13.1. Message: ConnectStatusRequest

Inherits: MeshRequest

Request status information for a pending request posted previously.

AccountID: String (Optional) Account identifier for which pending connection information is requested.

DeviceID: String (Optional) Device identifier of device requesting status information.

3.13.2. Message: ConnectStatusResponse

Reports the success or failure of a ConnectStatus transaction.

Inherits: MeshRequest The signed ConnectionResult object.

Result: SignedConnectionResult (Optional)

3.14. Transaction:

ConnectPending

Request: ConnectPendingRequest

Response: ConnectPendingResponse Request a list of pending requests for an administration profile.

3.14.1. Message: ConnectPendingRequest

Inherits: MeshRequest

Specify the criteria for pending requests.

AccountID: String (Optional) The account identifier of the account for which pending connection requests are requested.

SearchConstraints: SearchConstraints (Optional) Constrain the search to a specific time interval and/or limit the number and/or total size of data records returned.

3.14.2. Message: ConnectPendingResponse

Reports the success or failure of a ConnectPending transaction.

Inherits: MeshRequest A list of pending requests satisfying the **Pending: SignedConnectionRequest [0..Many]** criteria set out in the request.

PageKey: String (Optional) If non-null, indicates that the number and/or size of the data records returned exceeds either the SearchConstraints specified in the request or internal server limits.

3.15. Transaction: ConnectComplete

Request: ConnectCompleteRequest

Response: ConnectCompleteResponse Post response to a pending connection request.

3.15.1. Message: ConnectCompleteRequest

Reports the success or failure of a ConnectComplete transaction.

Inherits: MeshRequest The connection result to be posted to the
Result: SignedConnectionResult (Optional) portal. The result MUST
be signed by a valid administration key for the Mesh profile.

AccountID: String (Optional) The account identifier to which the
connection result is posted.

3.15.2. Message: ConnectCompleteResponse

Inherits: MeshRequest
Reports the success or failure of a
ConnectComplete transaction.

[No fields]

3.16. Transaction: Transfer

Request: TransferRequest
Response: TransferResponse Perform a bulk transfer of the log
between the specified transaction
identifiers. Requires appropriate authorization

[Not currently implemented]

3.16.1. Message: TransferRequest

Request a bulk transfer of the log between the specified transaction
identifiers. Requires appropriate authorization

Inherits: MeshRequest Constrain the search to a specific time
SearchConstraints: SearchConstraints (Optional) interval and/or
limit the number and/or total size of data records returned.

3.16.2. Message: TransferResponse

Inherits: MeshResponse
Reports the success or failure of a Transfer
transaction. If successful, contains the list of Mesh records to be
transferred.

DataItems: DataItem [0..Many] List of mesh data records matching
the request.

PageKey: String (Optional) If non-null, indicates that the number
and/or size of the data records returned exceeds either the

SearchConstraints specified in the request or internal server limits.

4. Assets

4.1. Data

4.2. Credentials

4.3. Reputation

4.3.1. Outbound Messaging Abuse ()

5. Risks

5.1. Confidentiality

Is a regulatory requirement GDPR/HIPPA

5.1.1. Privacy

Stronger requirement, given data but with restrictions on use

Unintended use within an organization may put it in default

GDPR

HIPPA

5.2. Integrity

Modification of data enables control breaches

5.3. Availability

5.3.1. Data loss

Loss of the pictures of the kids at 5

5.3.2. Partial data survivability

Where they buried Aunt Agatha's jewelry but not where they buried Aunt Agatha.

5.4. Inbound Messaging Abuse (Spam)

6. Threats

6.1. End point Compromise

7. Controls

7.1. Cryptographic

7.1.1. Triple lock

7.1.1.1. Transport Security

Traffic analysis protection

7.1.1.2. Message Security

Access control

Authentication / Integrity

7.1.1.3. Data Level Security

Data Confidentiality

Non-Repudiation

7.1.2. Key Protection

Use of platform provided facilities to bind private keys in the Device profile to the device is highly desirable. Ideally, private keys should be protected against extraction by hardware techniques presenting a high degree of resistance.

7.1.2.1. Windows

Use encrypted key store

Preferably use BitLocker

7.1.2.2. OSX

Use Key Ring

7.1.2.3. iOS

Use ???

7.1.2.4. Linux

Use the DBUS mechanism

7.1.2.5. Android

Hope and prayers.

7.1.3. Key and Nonce Generation

Use strong mechanisms as described in RFC???

Use of key co-generation as described in part 8 is advised

7.1.4. Key Escrow and Recovery

Master profile keys should be escrowed

Escrow strategies for DARE should take account of the fact that users may want some but not all their data assets to survive them.

7.1.5. Profile Verification

Check that the device credential has been signed by an administration device and that the administration device was properly authorized by the master profile.

Device catalog **MUST** be signed by the admin device.

Future ? provide protection against rollback attacks.

7.1.6. Identity Validation

See the separate document on the trust model

7.1.7. Trust Broker Accountability

Cert transparency type techniques

7.2. Mesh Messaging

7.2.1. Ingress Control

Every message is subject to access control

Mesh Services should perform abuse filtering on inbound mail

Mesh Services **MUST** apply user specified ingress control as specified in their contacts catalog.

7.2.2. Egress Control

Some applications may require egress control

For example, classified environments

Mail too stupid to send

7.2.3. Security Signal

Confirmation messages requiring payments

Need Accountability

Need to know the source of the accountability assertions

Should be distinguished from sender controlled part of a message

7.2.3.1. Brand

If messages are being sent on behalf of a corporate entity, this should be signaled to both sender and receiver

Sender ? remind them that they are speaking on behalf of another party

Receiver ? establish who is speaking by the familiar technique.

7.2.4. Accountability

Authentication and consequences

8. Security Considerations

This document comprises the security considerations for the use and implementation of the Mathematical Mesh.

8.1. Integrity

8.1.1. DNS Spoofing

8.1.2. TLS Downgrade

8.1.3. TLS Service Impersonation

8.1.4. Request Replay Attack

8.1.5. Response Replay Attack

8.2. Confidentiality

8.2.1. Side Channel Attack

8.2.2. Session Key Leakage

9. IANA Considerations

All the IANA considerations for the Mesh documents are specified in this document

10. Acknowledgements

A list of people who have contributed to the design of the Mesh is presented in [[draft-hallambaker-mesh-architecture](#)].

11. Normative References

[draft-hallambaker-mesh-architecture]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part I: Architecture Guide", Work in Progress, Internet-Draft, draft-hallambaker-mesh-architecture-14, 27 July 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-architecture-14>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

12. Informative References

[draft-hallambaker-mesh-developer]

Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", Work in Progress, Internet-Draft, draft-hallambaker-mesh-developer-10, 27 July 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-developer-10>>.