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.

This document is also available online at http://mathmesh.com/Documents/draft-hallambaker-mesh-schema.html [1].

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 October 6, 2019.

Copyright Notice

Copyright (c) 2019 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ........................................... 4
2. Definitions ........................................... 4
   2.1. Requirements Language .......................... 4
   2.2. Defined Terms ................................ 4
   2.3. Related Specifications ......................... 4
   2.4. Implementation Status .......................... 5
3. Mesh Profiles ......................................... 5
   3.1. Master Profile .................................. 5
   3.2. Device Profile .................................. 5
   3.3. Mesh Profile .................................... 5
4. Data Model ............................................. 5
   4.1. Data Model ....................................... 5
       4.1.1. Objects ................................... 5
       4.1.2. Catalogs ................................ 6
       4.1.3. Spools ................................... 6
5. Catalog Entries ....................................... 6
   5.1. Device ........................................... 6
   5.2. Contact .......................................... 6
   5.3. Credential ....................................... 6
   5.4. Network .......................................... 6
   5.5. Bookmark ........................................ 6
   5.6. Task ............................................. 6
   5.7. Application ...................................... 6
       5.7.1. SSH ....................................... 6
       5.7.2. Mail ...................................... 7
6. Messages ............................................... 7
   6.1. Completion ...................................... 7
   6.2. Connection ...................................... 7
   6.3. Contact .......................................... 7
   6.4. Confirmation .................................... 7
7. Schema ................................................ 7
   7.1. Shared Classes .................................. 7
       7.1.1. Structure: PublicKey ......................... 7
   7.2. Mesh Profile Objects ............................. 8
       7.2.1. Structure: Profile .......................... 8
       7.2.2. Keyset Classes ............................. 8
       7.2.3. Structure: EscrowedKeySet ................... 8
       7.2.4. Profile Classes ............................ 8
       7.2.5. Structure: ProfileMaster .................... 8
       7.2.6. Structure: ProfileDevice .................... 9
       7.2.7. Structure: ProfileApplication ............... 9
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.8</td>
<td>Structure: ProfileMesh</td>
<td>9</td>
</tr>
<tr>
<td>7.2.9</td>
<td>Structure: ProfileMeshDevicePublic</td>
<td>10</td>
</tr>
<tr>
<td>7.2.10</td>
<td>Structure: ProfileMeshDevicePrivate</td>
<td>10</td>
</tr>
<tr>
<td>7.2.11</td>
<td>Structure: DeviceRecryptionKey</td>
<td>10</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Structure: ProfileMeshDevicePrivate</td>
<td>10</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Structure: DeviceRecryptionKey</td>
<td>10</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Structure: Permission</td>
<td>10</td>
</tr>
<tr>
<td>7.3.4</td>
<td>Structure: Contact</td>
<td>11</td>
</tr>
<tr>
<td>7.3.5</td>
<td>Structure: Role</td>
<td>12</td>
</tr>
<tr>
<td>7.3.6</td>
<td>Structure: Address</td>
<td>12</td>
</tr>
<tr>
<td>7.3.7</td>
<td>Structure: Reference</td>
<td>12</td>
</tr>
<tr>
<td>7.3.8</td>
<td>Structure: Reference</td>
<td>13</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Structure: Permission</td>
<td>13</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Structure: Contact</td>
<td>13</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Structure: Role</td>
<td>13</td>
</tr>
<tr>
<td>7.4.4</td>
<td>Structure: Address</td>
<td>13</td>
</tr>
<tr>
<td>7.4.5</td>
<td>Structure: Reference</td>
<td>13</td>
</tr>
<tr>
<td>7.4.6</td>
<td>Structure: CatalogEntry</td>
<td>13</td>
</tr>
<tr>
<td>7.4.7</td>
<td>Structure: CatalogEntryDevice</td>
<td>13</td>
</tr>
<tr>
<td>7.4.8</td>
<td>Structure: CatalogEntryNetwork</td>
<td>14</td>
</tr>
<tr>
<td>7.4.9</td>
<td>Structure: CatalogEntryCredential</td>
<td>14</td>
</tr>
<tr>
<td>7.4.10</td>
<td>Structure: CatalogEntryContact</td>
<td>14</td>
</tr>
<tr>
<td>7.4.11</td>
<td>Structure: CatalogEntryContactRecryption</td>
<td>15</td>
</tr>
<tr>
<td>7.4.12</td>
<td>Structure: CatalogEntryBookmark</td>
<td>15</td>
</tr>
<tr>
<td>7.4.13</td>
<td>Structure: CatalogEntryTask</td>
<td>15</td>
</tr>
<tr>
<td>7.4.14</td>
<td>Structure: CatalogEntrySSH</td>
<td>15</td>
</tr>
<tr>
<td>7.4.15</td>
<td>Structure: CatalogEntryMail</td>
<td>15</td>
</tr>
<tr>
<td>7.4.16</td>
<td>Structure: CatalogEntryNetwork</td>
<td>15</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Structure: MeshMessage</td>
<td>17</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Structure: MeshMessageComplete</td>
<td>17</td>
</tr>
<tr>
<td>7.5.3</td>
<td>Structure: MessageConnectionRequest</td>
<td>18</td>
</tr>
<tr>
<td>7.5.4</td>
<td>Structure: MessageConnectionPIN</td>
<td>18</td>
</tr>
<tr>
<td>7.5.5</td>
<td>Structure: MessageContactRequest</td>
<td>19</td>
</tr>
<tr>
<td>7.5.6</td>
<td>Structure: MessageConfirmationRequest</td>
<td>19</td>
</tr>
<tr>
<td>7.5.7</td>
<td>Structure: MessageConfirmationResponse</td>
<td>19</td>
</tr>
<tr>
<td>7.5.8</td>
<td>Structure: MessageTaskRequest</td>
<td>19</td>
</tr>
<tr>
<td>8.</td>
<td>Security Considerations</td>
<td>19</td>
</tr>
<tr>
<td>9.</td>
<td>IANA Considerations</td>
<td>19</td>
</tr>
<tr>
<td>10.</td>
<td>Acknowledgements</td>
<td>20</td>
</tr>
<tr>
<td>11.</td>
<td>References</td>
<td>20</td>
</tr>
<tr>
<td>11.1.</td>
<td>Normative References</td>
<td>20</td>
</tr>
<tr>
<td>11.2.</td>
<td>Informative References</td>
<td>20</td>
</tr>
<tr>
<td>11.3.</td>
<td>URIs</td>
<td>20</td>
</tr>
<tr>
<td>Author’s Address</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>
1. Introduction

This document describes the data structures of the Mathematical Mesh with illustrative examples. For an overview of the Mesh objectives and architecture, consult the accompanying Architecture Guide [draft-hallambaker-mesh-architecture]. For information on the implementation of the Mesh Service protocol, consult the accompanying Protocol Reference [draft-hallambaker-mesh-protocol].

This document has two main sections. The first section presents examples of the Mesh profile, catalog entry and messages in use. The second section contains the schema reference. All the material in both sections is generated from the Mesh reference implementation [draft-hallambaker-mesh-developer].

Although some of the services described in this document could be used to replace existing Internet protocols including FTP and SMTP, the principal value of any communication protocol lies in the size of the audience it allows them to communicate with. Thus, while the Mesh Messaging service is designed to support efficient and reliable transfer of messages ranging in size from a few bytes to multiple terabytes, the near-term applications of these services will be to applications that are not adequately supported by existing protocols if at all.

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].

3. Mesh Profiles

Mesh profiles are signed assertions that describe a set of cryptographic credentials belonging to a user, a device or an account.

Profiles perform a similar role to X.509v3 certificates but with important differences:

- Profiles describe credentials, they do not make identity statements
- Profiles do not expire, there is therefore no need to support renewal processing.
- Profiles may be modified over time, the current and past status of a profile being recorded in an append only log.

3.1. Master Profile

A Mesh master profile provides the root of trust for a mesh user.

Unless exceptional circumstances require, a

```` Example SchemaMaster ````

3.2. Device Profile

```` Example SchemaDevice ````

3.3. Mesh Profile

```` Example SchemaMesh ````

4. Data Model

4.1. Data Model

4.1.1. Objects

Unique identifier

Append only log
Log can be purged.

4.1.2. Catalogs

Set of entries

Entry state machine (Add-Update*-Delete)*

4.1.3. Spools

Queue of messages

Message state machine (Post-(Read-Unread)*-Delete)

5. Catalog Entries

5.1. Device

```` Example SchemaEntryDevice ````

5.2. Contact

```` Example SchemaEntryContact ````

5.3. Credential

```` Example SchemaEntryCredential ````

5.4. Network

```` Example SchemaEntryNetwork ````

5.5. Bookmark

```` Example SchemaEntryBookmark ````

5.6. Task

```` Example SchemaEntryTask ````

5.7. Application

5.7.1. SSH

```` Example SchemaEntrySSH ```
5.7.2. Mail

```` Example SchemaEntryMail ````

6. Messages

All communications between Mesh accounts takes the form of a Mesh Message.

6.1. Completion

```` Example SchemaMessageCompletion ````

6.2. Connection

```` Example SchemaMessageConnection ````

6.3. Contact

```` Example SchemaMessageContact ````

6.4. Confirmation

```` Example SchemaMessageConfirmation ````

7. Schema

7.1. Shared Classes

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

7.1.1. Structure: PublicKey

The PublicKey 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
X509CSR: Binary (Optional)  X.509 Certificate Signing Request.
7.2. Mesh Profile Objects

Base class for all Mesh Profile objects.

7.2.1. Structure: Profile

Parent class from which all profile types 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.

7.2.2. Keyset Classes

7.2.3. Structure: EscrowedKeySet

A set of escrowed keys.

[No fields]

7.2.4. Profile Classes

7.2.5. Structure: ProfileMaster

Inherits: Profile

Describes the long term parameters associated with a personal profile.

This profile MUST be signed by

MasterSignatureKey: PublicKey (Optional)  The root of trust for the Personal PKI, the public key of the PMSK is presented as a self-signed X.509v3 certificate with Certificate Signing use enabled. The PMSK is used to sign certificates for the PMEK, POSK and PKEK keys.

MasterEscrowKeys: PublicKey [0..Many]  A Personal Profile MAY contain one or more PMEK keys to enable escrow of private keys used for stored data.
OnlineSignatureKeys: PublicKey [0..Many]  A Personal profile contains at least one OSK which is used to sign device administration application profiles.

7.2.6. Structure: ProfileDevice

Inherits: Profile

Describes a mesh device.

This profile MUST be signed by the DeviceSignatureKey

Description: String (Optional)  Description of the device

DeviceSignatureKey: PublicKey (Optional)  Key used to sign certificates for the DAK and DEK. The fingerprint of the DSK is the UniqueID of the Device Profile

DeviceAuthenticationKey: PublicKey (Optional)  Key used to authenticate requests made by the device.

DeviceEncryptionKey: PublicKey (Optional)  Key used to pass encrypted data to the device such as a DeviceUseEntry

7.2.7. Structure: ProfileApplication

Inherits: Profile

Contains the public description of a Mesh application.

[No fields]

7.2.8. Structure: ProfileMesh

Inherits: ProfileApplication

Contains the binding of a device to a MasterProfile. Each device has a separate profile which MUST be signed by an OnlineSignatureKey

Account: String (Optional)  Account address.

MasterProfile: DareMessage (Optional)  Master profile of the account being registered.

AccountEncryptionKey: PublicKey (Optional)  Key used to encrypt data under this profile
7.2.9. Structure: ProfileMeshDevicePublic

    Inherits: ProfileApplication

    DeviceProfile: DareMessage (Optional)  Device profile of the device making the request.

    Permissions: Permission [0..Many]  List of the permissions that the device has been granted.

7.2.10. Structure: ProfileMeshDevicePrivate

    Inherits: ProfileApplication

    Permissions: Permission [0..Many]  List of the permissions that the device has been granted.

    ProfileNonce: Binary (Optional)  Random nonce used to mask the fingerprint of the profile UDF.

    ProfileWitness: Binary (Optional)  Witness value calculated over the ProfileNonce and profile UDF

7.2.11. Structure: DeviceRecryptionKey

    UDF: String (Optional)  The fingerprint of the encryption key

    RecryptionKey: PublicKey (Optional)  The recryption key

    DeviceRecryptionKeyEncrypted: DareMessage (Optional)  The decryption key encrypted under the user’s device key.

7.3. Common Structures

7.3.1. Structure: Permission

    Name: String (Optional)

    Role: String (Optional)
Capabilities: DareMessage (Optional) Keys or key contributions enabling the operation to be performed

7.3.2. Structure: Contact

- Identifier: String (Optional)
- Account: String (Optional)
- FullName: String (Optional)
- Title: String (Optional)
- First: String (Optional)
- Middle: String (Optional)
- Last: String (Optional)
- Suffix: String (Optional)
- Labels: String [0..Many]
- Addresses: Address [0..Many]
- Locations: Location [0..Many]
Roles: Role [0..Many]

7.3.3. Structure: Role

  CompanyName: String (Optional)
  Addresses: Address [0..Many]
  Locations: Location [0..Many]

7.3.4. Structure: Address

  URI: String (Optional)
  Labels: String [0..Many]

7.3.5. Structure: Location

  Appartment: String (Optional)
  Street: String (Optional)
  District: String (Optional)
  Locality: String (Optional)
  County: String (Optional)
  Postcode: String (Optional)
7.3.6. 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.

7.4. Catalog Entries

7.4.1. Structure: CatalogEntry

[No fields]

7.4.2. Structure: CatalogEntryDevice

Inherits: CatalogEntry

Public device entry, indexed under the device ID

Account: String (Optional)  The Account to which this entry binds this device.

UDF: String (Optional)  UDF of the signature key

AuthUDF: String (Optional)  UDF of the authentication ID

ProfileMeshDevicePublicSigned: DareMessage (Optional)  The device profile

ProfileMeshDevicePrivateEncrypted: DareMessage (Optional)  The device profile

DeviceRecryptionKeys: DeviceRecryptionKey [0..Many]  Decryption key entries.

7.4.3. Structure: CatalogEntryCredential

Inherits: CatalogEntry

Inherits: CatalogEntry

Protocol: String (Optional)
7.4.4. Structure: CatalogEntryNetwork

Inherits: CatalogEntry
Inherits: CatalogEntry
Protocol: String (Optional)
Protocol: String (Optional)
Service: String (Optional)
Service: String (Optional)
Username: String (Optional)
Username: String (Optional)
Password: String (Optional)

7.4.5. Structure: CatalogEntryContact

Inherits: CatalogEntry
Inherits: CatalogEntry
Key: String (Optional)  Unique key.
Permissions: Permission [0..Many]  List of the permissions that the contact has been granted.
Contact: DareMessage (Optional)  The (signed) contact data.
7.4.6. Structure: CatalogEntryContactRecryption

Inherits: CatalogEntryContact

[No fields]

7.4.7. Structure: CatalogEntryBookmark

Inherits: CatalogEntry
Inherits: CatalogEntry
Uri: String (Optional)
Uri: String (Optional)
Title: String (Optional)
Title: String (Optional)
Path: String (Optional)

7.4.8. Structure: CatalogEntryTask

Inherits: CatalogEntry
Inherits: CatalogEntry
Task: DareMessage (Optional)
Task: DareMessage (Optional)
Key: String (Optional)  Unique key.

7.4.9. Structure: Task

Key: String (Optional)  Unique key.
Start: DateTime (Optional)
Start: DateTime (Optional)
Finish: DateTime (Optional)
Finish: DateTime (Optional)
StartTravel: String (Optional)
StartTravel: String (Optional)
FinishTravel: String (Optional)
FinishTravel: String (Optional)
TimeZone: String (Optional)
TimeZone: String (Optional)
Title: String (Optional)
Title: String (Optional)
Description: String (Optional)
Description: String (Optional)
Location: String (Optional)
Location: String (Optional)
Trigger: String [0..Many]
Trigger: String [0..Many]
Conference: String [0..Many]
Conference: String [0..Many]
Repeat: String (Optional)
Repeat: String (Optional)
Busy: Boolean (Optional)

7.4.10. Structure: CatalogEntryApplication

Inherits: CatalogEntry
Inherits: CatalogEntry
Key: String (Optional)
7.4.11. Structure: CatalogEntryApplicationEntry
    [No fields]

7.4.12. Structure: CatalogEntryApplicationRecryption
    [No fields]

7.4.13. Structure: CatalogEntryApplicationSSH
    [No fields]

7.4.14. Structure: CatalogEntryApplicationMail
    [No fields]

7.4.15. Structure: CatalogEntryApplicationNetwork
    [No fields]

7.5. Messages

7.5.1. Structure: MeshMessage
    MessageID: String (Optional)
    MessageID: String (Optional)
    Sender: String (Optional)
    Sender: String (Optional)
    Recipient: String (Optional)
    Recipient: String (Optional)
    References: Reference [0..Many]

7.5.2. Structure: MeshMessageComplete
    Inherits: MeshMessage
    [No fields]
7.5.3. Structure: MessageConnectionRequest

Inherits: MeshMessage

Inherits: MeshMessage

Account: String (Optional)

Account: String (Optional)

DeviceProfile: DareMessage (Optional)  Device profile of the device making the request.

ClientNonce: Binary (Optional)

ClientNonce: Binary (Optional)

ServerNonce: Binary (Optional)

ServerNonce: Binary (Optional)

Witness: String (Optional)

Witness: String (Optional)

PinID: String (Optional)  Pin identifier used to identify a PIN authenticated request.

7.5.4. Structure: MessageConnectionPIN

Inherits: MeshMessage

Inherits: MeshMessage

Account: String (Optional)

Account: String (Optional)

Expires: DateTime (Optional)

Expires: DateTime (Optional)

PIN: String (Optional)
7.5.5. Structure: MessageContactRequest

    Inherits: MeshMessage

    Inherits: MeshMessage

    Contact: DareMessage (Optional)  The contact data.

7.5.6. Structure: MessageConfirmationRequest

    Inherits: MeshMessage

    Inherits: MeshMessage

    Text: String (Optional)

7.5.7. Structure: MessageConfirmationResponse

    Inherits: MeshMessage

    Inherits: MeshMessage

    ResponseID: String (Optional)

    ResponseID: String (Optional)

    Accept: Boolean (Optional)

7.5.8. Structure: MessageTaskRequest

    Inherits: MeshMessage

    [No fields]

8. Security Considerations

    The security considerations for use and implementation of Mesh services and applications are described in the Mesh Security Considerations guide [draft-hallambaker-mesh-security].

9. IANA Considerations

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

11. References

11.1. Normative References

[draft-hallambaker-mesh-architecture]

[draft-hallambaker-mesh-protocol]
"[Reference Not Found!]".

[draft-hallambaker-mesh-security]
"[Reference Not Found!]".


11.2. Informative References

[draft-hallambaker-mesh-developer]

11.3. URIs


Author’s Address

Phillip Hallam-Baker

Email: phill@hallambaker.com